Technical Information

STARDOM 計装テクニカルガイド

TI 34P02A03-01

A 編　概説（STARDOM 計装）

B 編　自律型コントローラ FCN/FCJ

C 編　VDS（Web ベース HMI/SCADA ソフトウェア ASTMAC VDS）

D 編　NPAS POU（PAS ポートフォリオ概説）

E 編　FCN/FCJ 設置仕様

F 編　仕様（GS 集）

F1.　STARDOM 概要

F2.　FCN/FCJ

F3.　APPF

F4.　VDS

F5.　STARDOM ガイド

YOKOGAWA
横河電機株式会社

TI 34P02A03-01
2006.06.20　初版（YK）
2014.07.31　10 版（YK）

はじめに

■ 本書について

STARDOM 計装は、以下を実現しています。

- 高い信頼性
- エンジニアリングが簡単、高品質
- 他システムとも簡単に接続できるオープンなシステム
- 長期的な運用も安心サポート

本書では、この STARDOM 計装の機能について、詳しく説明しています。

■ 記載内容について

本書の記載内容は、本書発行時点のハードウェアのスタイル、ソフトウェアのリリースの仕様に基づいています。
稼働中のハードウェアとソフトウェアの組み合わせにより、機能が制限されることがあります。下記ホームページでご確認ください。

http://STARDOM.jp
→会員サイト　Yokogawa Plus
ネットワークベース生産システム（STARDOM）
リリース別（OS、 CPU、 Web）対応表
STARDOM コンポーネント互換性

■ 本書の目的と対象とする読者

本書は、機種選定用の見積ガイドとして STARDOM 計装のエンジニアリング、見積をされる方を対象に制作されています。

10th Edition : July 2014 (YK)
All Rights Reserved、 Copyright © 2006. Yokogawa Electric Corporation

STARDOM 計装
テクニカルガイド
Part-B　自律型コントローラ FCN/FCJ 編

TI 34P02A03-01　10 版

目次　Part-B 自律型コントローラ FCN/FCJ 編 i

B 1.　自律型コントローラFCN/FCJの特長 B-1

B 1.1　モジュール実装タイプ「FCN」 B-3

B 1.2　オールインワンタイプ「FCJ」 B-6

B 2.　FCN/FCJ機能概要 B-7

B 2.1　FCN/FCJの制御機能 B-7

B 2.2　FCN/FCJと接続可能な機器 B-9

B 2.2.1 VDSとの接続 B-11

B 2.2.2 OPCサーバによる汎用SCADAとの接続 B-13

B 2.2.3 CENTUM接続（データ統合） B-15

B 2.2.4 CENTUM接続（オペレーション統合） B-18

B 2.2.5 他のFCN/FCJ、各種PLC、Modbus通信機器との接続 B-20

B 2.2.6 表示器との接続 B-21

B 2.2.7 各種フィールド機器との接続 B-21

B 2.3　FCN/FCJの自律発信機能 B-22

B 2.3.1 情報発信パッケージInfoWell B-23

B 2.3.2 Javaアプリケーション B-34

B 2.3.3 時刻同期機能 B-38

B 2.4　制御ネットワーク B-41

B 2.4.1 制御ネットワークの二重化 B-42

B 2.4.2 制御ネットワークの2系統化 B-42

B 3.　FCN/FCJハードウェア B-43

B 3.1　FCNモジュール構成 B-43

B 3.1.1 ベースモジュールNFBU200/NFBU050 ..... B-46

B 3.1.2 電源モジュール（FCN用） B-48

B 3.1.3 CPUモジュール（FCN用） B-50

B 3.1.4 SBバスリピートモジュール（FCN用） B-54

B 3.1.5 I/Oモジュール B-56

B 3.2　自律型コントローラFCJ B-61

B 4.　FCN/FCJのソフトウェア B-62

B 4.1　開発環境／ツール B-63

B 4.1.1 リソースコンフィギュレータ B-65

B 4.1.2 ロジックデザイナ B-67

B 4.1.3 ファンクション／ファンクションブロック B-74

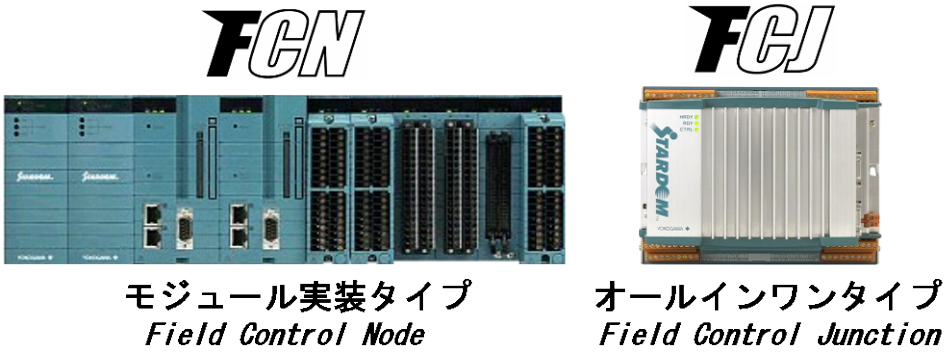
B 4.1.4 アプリケーションポートフォリオ（APPF） B-81
B 4.1.5 FCN/FCJ Javaアプリケーション開発 B-82
B 4.1.6 WebブラウザによるFCN/FCJの設定 B-84
B 4.2 FCN/FCJシステム構築手順 B-86
B 4.2.1 ハードウェアのセットアップ B-87
B 4.2.2 FCN/FCJ初期設定手順 B-91
B 4.2.3 制御アプリケーションの作成手順 B-93
B 4.2.4 制御アプリケーション作成上の注意 B-95
B 4.3 制御アプリケーション B-96
B 4.3.1 制御アプリケーションの構成 B-97
B 4.3.2 制御アプリケーションのダウンロード B-102
B 4.3.3 タスクのスケジュール B-104
B 4.3.4 入出力処理 B-111
B 4.3.5 保持変数（リテインデータ） B-113
B 4.4 機能仕様 B-117
B 4.5 パフォーマンス B-118
B 4.5.1 FCN/FCJのスキャン周期と制御周期 B-118
B 4.5.2 制御アプリケーション容量の算出 B-119
B 4.5.3 パフォーマンスの確認 B-121
B 4.5.4 入出力処理のパフォーマンス B-123
B 5. 製品一覧 B-125
B 5.1 ソフトウェア製品一覧 B-125
B 5.2 アプリケーション開発に必要なソフトウェア B-127
B 5.3 FCN/FCJ動作に必要なソフトウェア B-128
B 5.4 ハードウェア製品一覧 B-129
B 6. 保守 B-131
B 6.1 全データのバックアップ／リストア（R3.30以降） B-131
B 6.1.1 全データのバックアップ B-131
B 6.1.2 全データのリストア B-136
B 6.2 メンテナンスメニュー B-140
B 6.2.1 システム設定ファイル B-141
B 6.3 オンラインダウンロード B-142
B 6.3.1 オンラインダウンロード機能概要 B-142
B 6.4 FCN/FCJハードウェアの保守 B-143
B 6.4.1 FCN保守・交換作業上注意事項 B-143
B 6.4.2 FCN各モジュールの交換 B-144
B 6.4.3 FCJ故障時の交換 B-147
B 6.5 有寿命部品 B-148
B 6.5.1 FCN CPUモジュール／FCJのバッテリ B-148
B 6.5.2 FCN電源モジュール B-150
B 6.5.3 FCN I/Oモジュール（NFDR541） B-150

B 7. その他 B-151
B 7.1 ライセンス B-151
B 7.1.1 ライセンスの種類 B-151
B 7.1.2 キーコード B-152
B 7.1.3 ライセンスID B-152
B 7.1.4 ライセンスの発行 B-152
B 7.2 CPUモジュール（FCN）二重化時の動作 B-154
B 7.2.1 CPUモジュール二重化時の制御権 B-156
B 7.2.2 All Program Copy（APC） B-157
B 7.2.3 制御権情報の初期化方法 B-162
B 7.2.4 注意事項 B-163
B 7.3 ネットワークの二重化 B-165
B 7.3.1 STARDOMネットワーク二重化機能 B-169
B 7.3.2 アプリケーションによる通信の二重化 B-181
B 7.4 通停電時の動作 B-183
B 7.4.1 停電不感時間 B-183
B 7.4.2 瞬停時／停電時のFCNの動作 B-183
B 7.4.3 停電復帰後の動作 B-184
B 7.5 障害発生時の動作 B-185
B 7.5.1 自己診断 B-185
B 7.5.2 動作状態 B-186

Blank Page

B1. 自律型コントローラ FCN/FCJ の特長

自律型コントローラ FCN（Field Control Node）/FCJ（Field Control Junction）は、連続制御とシーケンス制御の融合といった従来の機能に加え、高分散・高信頼・部品化といった特長のあるアーキテクチャを備え、新たな生産制御システムを実現します。用途や規模に応じて、モジュール実装タイプ FCN とオールインワンタイプ FCJ の 2 種類を用意しています。



モジュール実装タイプ
Field Control Node

オールインワンタイプ
Field Control Junction

図　自律型コントローラ FCN/FCJ

● 高分散

コンパクトな構造と HMI ソフトウェアに依存しない独立性の高いアーキテクチャにより、分散配置を指向します。これにより、導入から増改造まで迅速な対応を可能とする「機能分散」を実現すると共に、障害による影響を最小限にとどめる「危険分散」や遠隔地であっても自律的な運転が可能な「広域分散」を実現します。

● 高信頼

ECC 機能付きメモリや RAS 機能の装備など堅牢なハードウェアが安定操業を提供します。また、重要な設備にも適用できるように FCN では、電源、CPU、制御ネットワーク、内部バス（SB バス）の二重化、各モジュールの活線挿抜が可能です。

● 部品化

制御ロジックの開発言語は国際標準規格 IEC 61131-3 に準拠しています。アプリケーションプログラムの階層構造や部品化により作成したプログラムの再利用を推進します。
また、当社が長年培った制御アプリケーションノウハウをライブラリ化した「アプリケーションポートフォリオ（APPF）」を用意しています。必要な制御アプリケーション機能部品の利用によって高度な機能が容易に実現でき、かつ安定した動作を約束します。

■ 自律型コントローラ FCN/FCJ のアーキテクチャ

PLC の内部構造と FCN/FCJ の内部構造の差を下図に示します。FCN/FCJ の特長は、工業用 PC と同等の CPU の採用、制御プログラムと情報処理プログラムの並列処理実行です。これらのプログラムは、リアルタイム OS の上で実行されます。また I/O モジュールとの接続も高速二重化内部バス（SB バス）を用いて高信頼かつ高速に実行されます。さらに CPU 内蔵のネットワーク接続機能により、SCADA や他のコントローラと容易に接続できます。
このように、ひとつの CPU にネットワーク機能、情報処理機能、そして制御機能をまとめてあるので、単なる PLC としての制御機能だけでなく、情報発信機能や情報処理機能を統合した形でスマートかつ安価に提供できます。

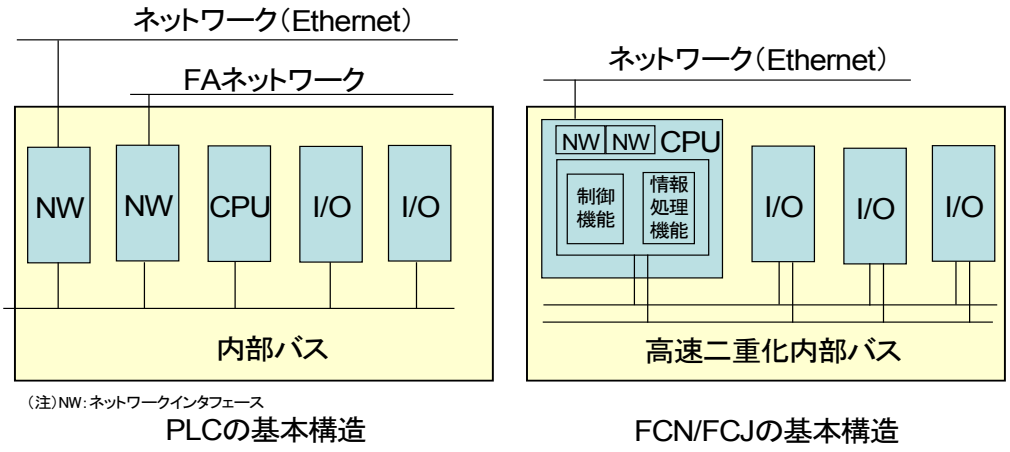


図　PLC の基本構成と自律型コントローラ FCN/FCJ の基本構成

● 高信頼の二重化構成

高信頼の二重化構成は、下図のようにシンプルに構成されます。高速二重化内部バス（SB バス）を介して二重化制御のための情報が CPU 同士で交換されます。ネットワークの二重化も、CPU に 2 系統準備されたネットワークインタフェースを用いて容易に実現できます。

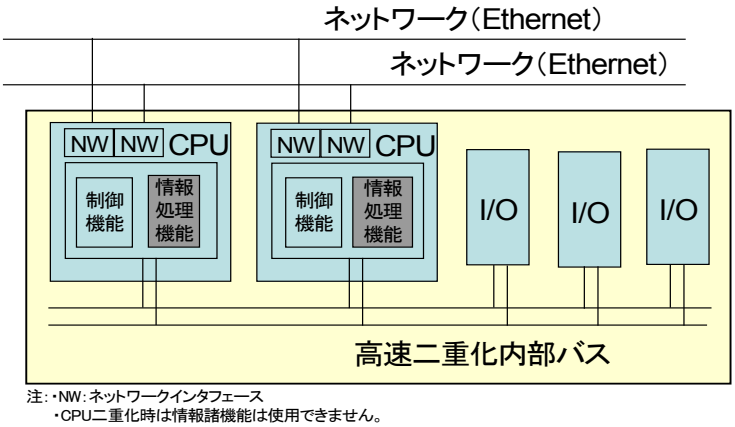


図　自律型コントローラ FCN/FCJ の二重化構成

B1.1　モジュール実装タイプ「FCN」

FCN は適用する設備の規模や信号の種類に応じて任意の I/O モジュールを組み合わせて使用するモジュール実装タイプのコントローラで、以下の特長があります。

■ FCN の特長

- 分散配置を実現する、現場設置が可能なコンパクトな構造
- ECC（Error Correcting Code）機能付きメモリ、充実した RAS（Reliability Availability Serviceability）機能の装備など堅牢な CPU モジュール
- 制御ネットワークの二重化が可能（単独に二重化にできます）
- 各モジュールの活線挿抜が可能
- 制御ネットワークは 100/10Mbps の汎用 Ethernet を使用
- CPU モジュールにシリアルポートを装備し表示器、電力モニタ、温調計などを接続可能

参照

表示器、電力モニタ、温調計などの接続の詳細については、「B2.2.5 他の FCN/FCJ、各種 PLC、Modbus 通信機器との接続」を参照してください。

● FCN（ベースモジュール NFBU200 使用時）の特長

- コンパクトな構造（W482.6×H132.5×D170.8 mm）(*1)
- 電源、CPU、SB バス（内部バス）の二重化が可能（各々単独に二重化にできます）
- I/O 拡張用に増設ユニットを 2 台まで接続可能

*1: 19 インチラック取り付け金具を含めた寸法（別途 DIN レール取り付けも可能）

参照

二重化の詳細については、「B3.1 FCN モジュール構成」を参照してください。

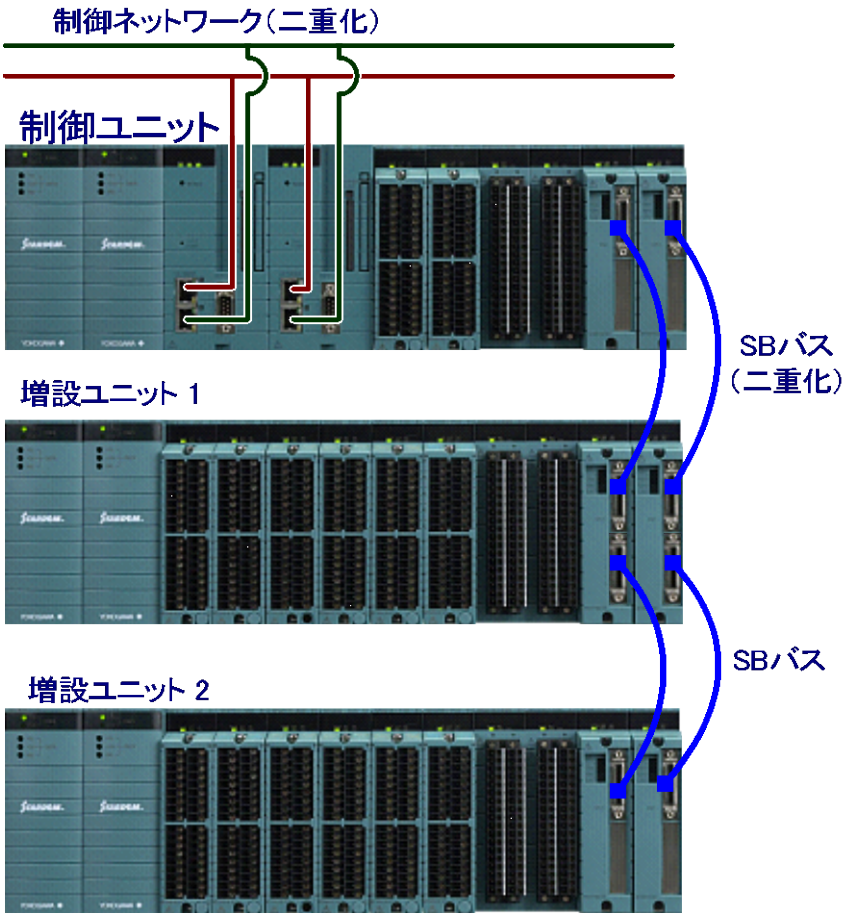


図　自律型コントローラ FCN（ベースモジュール NFBU200 使用時）

● FCN（ショートベースモジュール NFBU050 使用時）の特長

・ コンパクトな構造（W283×H131×D156.1 mm）

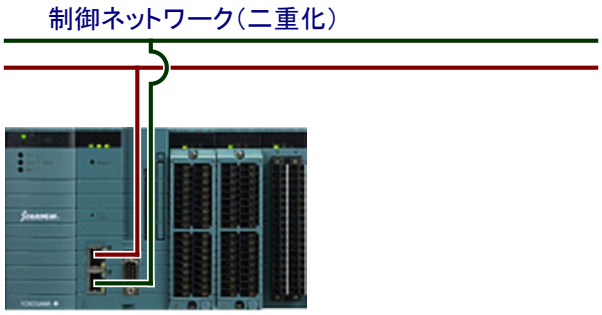


図　自律型コントローラ FCN（ショートベースモジュール NFBU050 使用時）

補足

ショートベースモジュール NFBU050 の SB バスはシングルです。

B1.2 オールインワンタイプ「FCJ」

FCJ は、本体に I/O インタフェースや通信ポートなど各種機能を内蔵したオールインワンタイプのコントローラで、以下の特長があります。

● FCJ の特長

- 装置組み込みや分散配置を実現するコンパクトかつ耐環境性に優れた構造（W213.8×H156×D95 mm (*1)）
- 下記 I/O を標準実装
  AI（1～5VDC）　：6 点
  AO（4～20mADC）　：2 点
  DI（定格入力 24VDC）　：16 点
  DO（定格負荷 24VDC (*2)）　：16 点
- ECC（Error Correcting Code）機能付きメモリ、充実した RAS（Reliability Availability Serviceability）機能の装備
- 制御ネットワークの二重化が可能（制御ネットワークは 100/10Mbps の汎用 Ethernet を使用）
- シリアルポート（RS-232-C×2 ポート）を装備し表示器、電力モニタ、温調計などを接続可能
- FOUNDATION fieldbus インタフェース（2 ポート）をオプションで実装可能 (*3)

*1:　取り付け金具は除きます。
*2:　外部に 24VDC の電源が必要です。
*3:　FOUNDATION fieldbus 機能を定義するツールは、STARDOM においては英文版のみです。

参照

シリアルポートへの表示器、電力モニタ、温調計などの接続の詳細については、「B2.2.5 他の FCN/FCJ、各種 PLC、Modbus 通信機器との接続」を参照してください。

図　FCJ

B2. FCN/FCJ 機能概要

B2.1 FCN/FCJ の制御機能

FCN/FCJ の制御機能は IEC 61131-3 で規定されるファンクション、ファンクションブロックや各種アプリケーションポートフォリオを用いて自由に構築できます。
各種アプリケーションフォリオや用いると、高度な制御機能を簡単かつ高品質に構築できます。

■ ファンクション、ファンクションブロック

ファンクション、ファンクションブロックには、以下のものがあります。

- 型変換ファンクション
- 数値ファンクション
- 算術ファンクション
- ブールビット演算ファンクション
- ビット文字列ファンクション
- 選択ファンクション
- 比較ファンクション
- 文字列ファンクション
- バイステーブルファンクションブロック
- エッジ検出型ファンクションブロック
- カウンタファンクションブロック
- タイマファンクションブロック
- ビット操作ファンクション
- FCN/FCJ 基本ファンクションブロック
- FCN/FCJ 間通信ファンクションブロック
- FA-M3 エミュレーション（シリアル通信）ファンクションブロック
- 通信ユーティリティファンクションブロック
- Ethernet 通信ファンクションブロック
- シリアル通信ファンクションブロック

参照

ファンクション、ファンクションブロックについては、「B4.1.3 ファンクション／ファンクションブロック」を参照してください。

■ PAS ポートフォリオ

プロセス制御を実現する以下の高機能部品を PAS ポートフォリオとして用意しています。

- 入出力データ処理 POU
- 連続制御 POU
- 数値演算 POU
- シーケンス制御 POU
- FF-H1（フィールドバス）POU
- ユーティリティ POU

参照

PAS ポートフォリオについては、「Part-D NPAS POU 編」を参照してください。

■ その他のアプリケーションポートフォリオ

各種機器との通信や特定アプリケーション向け高機能ポートフォリオも用意しています。

● 通信ポートフォリオ

PLC（FA-M3、MELSEC、SYSMAC）、Modbus 対応機器、DNP3 対応機器や横河製デジタル指示調節計との通信機能

● SAMA ポートフォリオ

電力 SAMA 仕様に準拠した部品

● AP 特化ポートフォリオ

アプリケーションに特化した機能を部品化（ボイラ制御ポートフォリオ、ボイラ補機制御ポートフォリオ）

参照

- 「FA-M3 通信ポートフォリオ」「MELSEC 通信ポートフォリオ」「SYSMAC 通信ポートフォリオ」「温調計通信ポートフォリオ」「SAMA ポートフォリオ」については、本書「F3. GS 集：APPF 関連」「FCN/FCJ アプリケーションポートフォリオ」（GS 34P02P20-01）を参照してください。
- 「Modbus 通信ポートフォリオ」については、本書「F3. GS 集：APPF 関連」「Modbus 通信ポートフォリオ」（GS 34P02P21-01）を参照してください。
- 「DNP3 通信ポートフォリオ」については、本書「F3. GS 集：APPF 関連」「DNP3 通信ポートフォリオ」（GS 34P02P22-01）を参照してください。

B2.2 FCN/FCJ と接続可能な機器

FCN/FCJ は、下表のような種々の機器と容易に接続でき、多種多様なシステムを構築できます

- **VDS/ASTMAC**
- **OPC サーバ**
- **FCN/FCJ**
- **各種 PLC、Modbus 通信機器**
- **表示器**
- **各種フィールド機器**

表　FCN/FCJ に接続可能な機器

分類	機器名称	インタフェース	対応パッケージ
自律型コントローラ	横河電機製　FCN/FCJ	Ethernet	標準対応
SCADA	横河電機製　VDS/ASTMAC	Ethernet	標準対応
	OPC 対応機器	Ethernet	FCN/FCJ OPC サーバ for Windows
DCS	横河電機製 CENTUM VP（UGS 経由） CENTUM CS3000（UGS 経由）	Ethernet	標準対応
	横河電機製 CENTUM VP（SIOS 経由） CENTUM CS3000（SIOS 経由）	Ethernet	FCN/FCJ OPC サーバ for Windows
	横河電機製 CENTUM VP（FCS 経由） CENTUM CS3000（FCS 経由）	Modbus（RS-232-C、RS-422/RS-485、Ethernet）	Modbus 通信ポートフォリオ
PLC	横河電機製　FA-M3	Ethernet、 RS-232-C、 RS-422/RS-485	FA-M3 通信ポートフォリオ
	三菱電機製　MELSEC	Ethernet	MELSEC 通信ポートフォリオ
	オムロン製　SYSMAC	RS-232-C、 RS-422/RS-485	SYSMAC 通信ポートフォリオ
電力モニタ	横河電機製 UPM100、 UPM101、 UZ005（S3 以降）、 PR201（S2 以降）、 PR300 UPM01、 UPM02、 UPM03	RS-232-C RS-422/RS-485	電力モニタ通信ポートフォリオ
温調計	横河電機製　Green シリーズ	RS-422/RS-485	温調計通信ポートフォリオ
	横河電機製 UTAdvanced シリーズ	Ethernet 、RS-422/RS-485	Modbus 通信ポートフォリオ
フィールド機器	FOUNDATION fieldbus H1 機器	FOUNDATION fieldbus H1	PAS ポートフォリオ
	HART 対応機器	HART	PAS ポートフォリオ
	CANopen 対応機器	CANopen	標準対応
	PROFIBUS-DP 対応機器	PROFIBUS-DP	標準対応
表示器	横河電機製 TOP2501、TOP2301 TOP3600T、TOP3301S	Modbus（RS-232-C、RS-422/RS-485、Ethernet）	Modbus 通信ポートフォリオ
	デジタル製 ・SimpleTouch ST400-AG41-24V、 ST401-AG41-24V ・GP シリーズ GP377 シリーズ、 GP77R シリーズ、 GP2000 シリーズ GP3000 シリーズ ・GLC シリーズ GLC2000 シリーズ ・Factory Gateway その他 Modbus 通信対応の表示器	Modbus（RS-232-C、RS-422/RS-485、Ethernet）	Modbus 通信ポートフォリオ

分類	機器名称	インタフェース	対応パッケージ
その他	Modbus 通信に対応した機器	Modbus（RS-232-C、RS-422/RS-485、Ethernet）	Modbus 通信ポートフォリオ
	DNP3 通信に対応した機器	DNP3（RS-232-C、RS-422/RS-485、Ethernet）	DNP3 通信ポートフォリオ

B2.2.1 VDS との接続

FCN/FCJ は、STARDOM を構成するコンポーネント製品である ASTMAC VDS（Web ベース HMI/SCADA ソフトウェア、以下 VDS と略す）に標準機能で接続できます。さらに、FCN/FCJ の各種データは、専用タグなどで簡単に VDS からアクセス可能です。

参照

・ VDS の詳細については、「PART-C VDS 編」を参照してください。

● 接続台数

FCN/FCJ と VDS は、制御ネットワーク（Ethernet）で接続されます。それぞれが接続可能な台数は下記のとおりです。

・ VDS
  VDS は、I/O 機器（FCN/FCJ を含む）最大 32 台と接続可能です。
・ FCN/FCJ
  FCN/FCJ は、VDS と FCN/FCJ OPC サーバの合計で最大 4 台と接続可能です。

● データアクセスが容易

VDS からアクセスしたい FCN/FCJ 上の変数は、変数定義の際に外部参照可能な変数と定義しておけば、VDS 上の監視タグ（コントロールオブジェクト）を容易に作成できます。さらに FCN/FCJ の各種データにアクセスするために、下表の専用コントロールオブジェクトも用意されています。

表　FCN/FCJ アクセス用コントロールオブジェクト

オブジェクト	機能
FCN/FCJ RAS オブジェクト	RAS 情報を VDS で表示するためのオブジェクト
FCN/FCJ メッセージオブジェクト	FCN/FCJ で発生したアラームを定周期で監視するオブジェクト
PAS POU オブジェクト	PAS ポートフォリオの各種データを収集するオブジェクト

参照

PAS ポートフォリオについては、「B4.1.4　アプリケーションポートフォリオ」を参照してください。

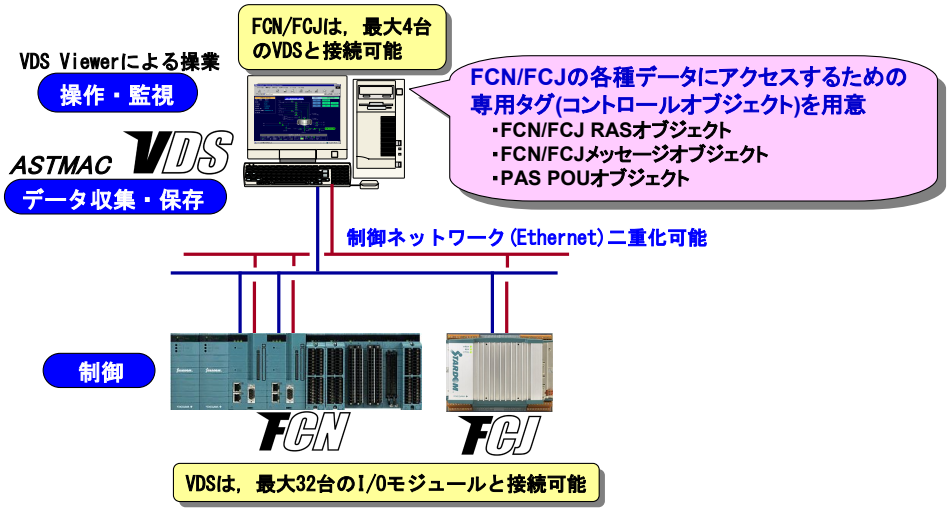


図　FCN/FCJ と VDS の接続

B2.2.2 OPC サーバによる汎用 SCADA との接続

オプションパッケージである「FCN/FCJ OPC サーバ for Windows」を用いれば、VB、VBA、VC++で作成したアプリケーションや、OPC DA 2.0 をサポートしている汎用 SCADA ソフトウェアにも、FCN/FCJ が接続可能になります。これにより、既存 HMI の資産を生かしつつ多くの特長を持った FCN/FCJ を採用できます。

● FCN/FCJ OPC サーバ for Windows 仕様

FCN/FCJ とは制御ネットワーク（Ethernet）で接続され、その接続台数は下表のとおりです。

表　FCN/FCJ OPC サーバ for Windows 概略仕様

項目		仕様
接続クライアント数		最大 100 クライアント
FCN/FCJ 接続台数		最大 100 台
データアクセス（DA）		OPC-F Data Access Custom Interface Standard Ver.2.05a 準拠
	グループオブジェクト数	最大 1、000 グループ
	登録アイテム ID 数	10、000／グループ　全グループ合計で 100、000
	データ更新周期	1、000～3、600、000 ms （1 秒～1 時間）
メッセージアクセス（A&E）		OPC-F Alarms and Events Custom Interface Standard Version 1.10 準拠 メッセージ通知／フィルタ (*1)
	イベント登録オブジェクト数 (*2)	1、000 オブジェクト
FCN/FCJ 二重化対応		CPU 二重化 FCN 接続可能、ネットワーク二重化可能 (*3)

*1:　「Ack」と「エリアのブラウズ」機能はサポートしていません。
*2:　イベントサブスクリプションオブジェクト数
*3:　ネットワークを二重化するには、別途「FCN/FCJ OPC サーバ用ネットワーク二重化プログラム」が必要です。

参照

FCN/FCJ OPC サーバ用ネットワーク二重化プログラムの詳細については、本書「F2. GS 集：FCN/FCJ 関連」「FCN/FCJ OPC サーバ用ネットワーク二重化プログラム」（GS 34P02Q62-01）を参照してください。

● 動作環境

表　FCN/FCJ OPC サーバ for Windows の動作環境

項目	仕様	
パソコン	PC/AT 互換機	
CPU	Windows 7（32 bit） Windows Server 2008（32 bit）	1 GHz 以上の 32 bit（x86）プロセッサ または 64 bit（x64）プロセッサ
	Windows 7（64 bit） Windows Server 2008 R2（64 bit）	2 GHz 以上の 64 bit（x64）プロセッサ
RAM	Windows 7（32 bit/64 bit） Windows Server 2008（32 bit） Windows Server 2008 R2（64 bit）	2 GB 以上
ハードディスク	20GB 以上の空き領域	
Ethernet アダプタ	100BASE-TX、10BASE-T または 10BASE5 のいずれか一つが必要 （下記 OS で動作するもの）	
DVD-ROM ドライブ	インストール時のみ必要（下記 OS で動作するもの）	
OS	Windows 7 Professional SP1 (32 bit/64 bit) Windows Server 2008 Standard Edition SP2 (32 bit) Windows Server 2008 R2 Standard Edition SP1 (64 bit)	

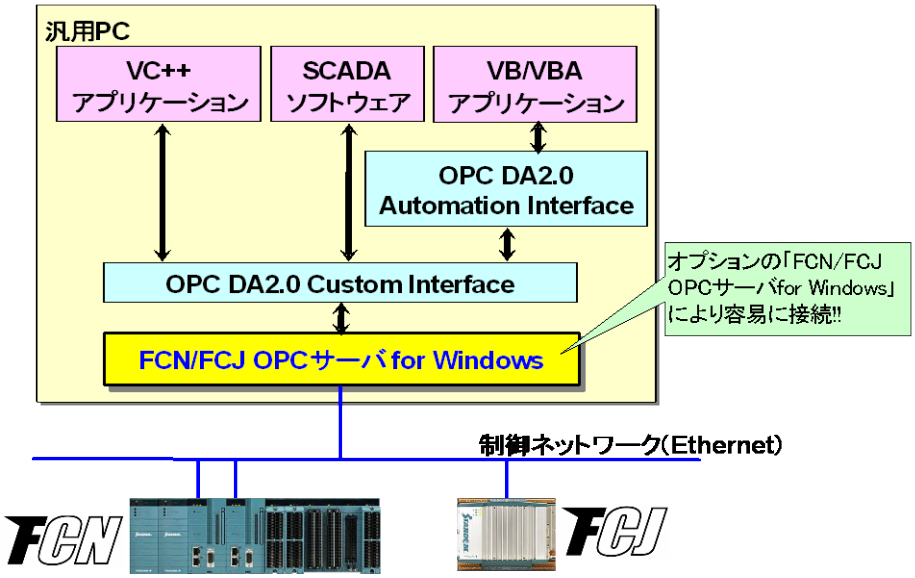


図　FCN/FCJ と汎用 SCADA、VB アプリケーションとの接続

参照

FCN/FCJ OPC サーバ for Windows の詳細については、本書「F2. GS 集：FCN/FCJ 関連」「FCN/FCJ OPC サーバ for Windows」（GS 34P02Q61-01）を参照してください。

B2.2.3 CENTUM 接続（データ統合）

CENTUM 接続（データ統合）は、CENTUM の FCS（フィールドコントロールステーション）のサブシステムとして FCN/FCJ を接続し、この FCN/FCJ のデータを、CENTUM のアナログ／デジタル入出力信号と同様に扱う手法です。一般的な PLC と同様に FCN/FCJ を CENTUM に接続できます。下記の 2 種類があります。

- **サブシステム接続**
- **汎用サブシステムゲートウェイ（GSGW）による接続**

■ サブシステム接続

CENTUM の FCS に FCN/FCJ をサブシステムとして接続します。
通信方式には下記の 2 種類があります。

- RTU mode
- Modbus/TCP

どちらも FCS の通信モジュールを用いて FCN/FCJ と通信します。サブシステムとして接続された FCN/FCJ のデータは、FCS のアナログ／デジタル入出力信号と同様に、機能ブロックの入出力端子に接続されて利用されます。
FCS と FCN/FCJ の通信は冗長化もできます。

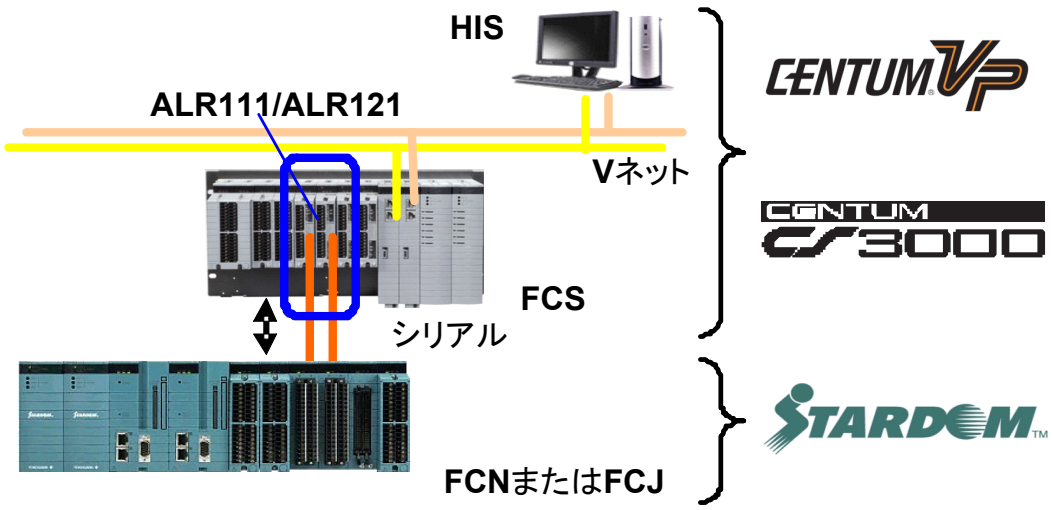


図　RTU mode による CENTUM 接続（データ統合）例

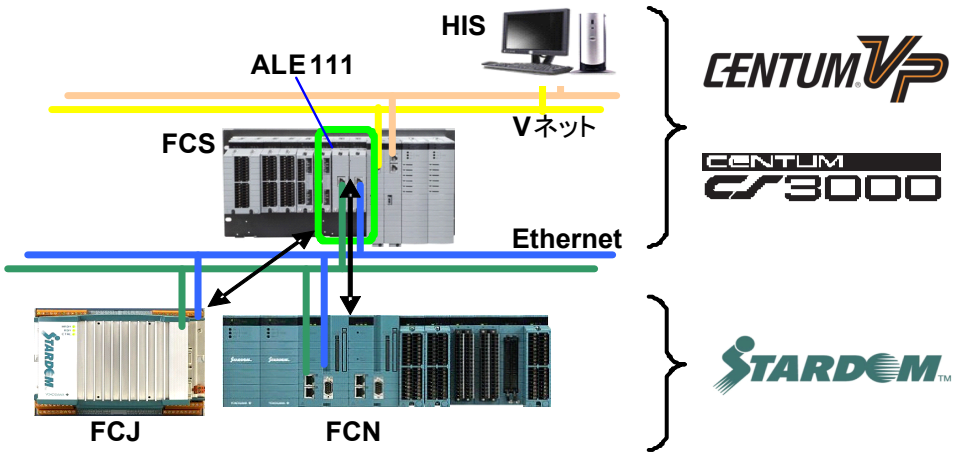


図　Modbus/TCP による CENTUM 接続（データ統合）例

表　CENTUM 接続（データ統合）：サブシステム接続

通信方式	システム	接続ポート［参照ドキュメント］		オプションソフトウェア［参照ドキュメント］	備考
RTU mode（シリアル通信）	CENTUM	ALR111、ALR121 CENTUM VP: ［GS 33M50G10-40］ CS 3000: ［GS 33Q06Q46-31］		Modbus 通信パッケージ（ALR111、ALR121 用） CENTUM VP: ［GS 33M15E10-40］ CS 3000: ［GS 33Q03L40-33］	通信の二重化も可能
	FCN/FCJ	FCN	NFLR111:RS-232-C、 NFLR121:RS-422/RS-485 ［GS 34P02Q12-01］ ［GS 34P02Q36-01］	NT8035J Modbus 通信ポートフォリオ ［GS 34P02P21-01］	
		FCJ	シリアルポート（2ch） ［GS 34P02Q11-01］		
Modbus/TCP（Ethernet 通信）	CENTUM	ALE111 CENTUM VP: ［GS 33M50G10-40］ CS 3000: ［GS 33Q06Q46-31］		Modbus 通信パッケージ（ALE111 用） CENTUM VP: ［GS 33M15E20-40］ CS 3000: ［GS 33Q03L40-34］	通信の二重化も可能
	FCN/FCJ	FCN	ネットワークポート（100BASE-TX） ［GS 34P02Q12-01］	NT8035J Modbus 通信ポートフォリオ ［GS 34P02P21-01］	
		FCJ	ネットワークポート（100BASE-TX） ［GS 34P02Q11-01］		

■ 汎用サブシステムゲートウェイ（GSGW）による接続

汎用サブシステムゲートウェイ（GSGW）は、サブシステムを操作監視するための CENTUM のステーションです。
汎用 PC をプラットフォームとして用い、OPC Foundation が規定した汎用の OPC DA（OPC：OLE for Process Control）インタフェースにより FCN/FCJ 側の OPC サーバ（FCN/FCJ OPC サーバ for Windows、以降 FCN/FCJ OPC サーバ）が搭載された汎用 PC を経由してサブシステム（FCN/FCJ）と通信します。
サブシステム（FCN/FCJ）のデータは、GSGW 機能ブロックの入出力端子に割り付けられます。この GSGW の機能ブロックは、FCS と同様に HIS から操作監視ができます。ただし、GSGW はサブシステムのデータ監視が主な目的です。PID ブロックなどの制御を目的とした機能ブロックはありません。

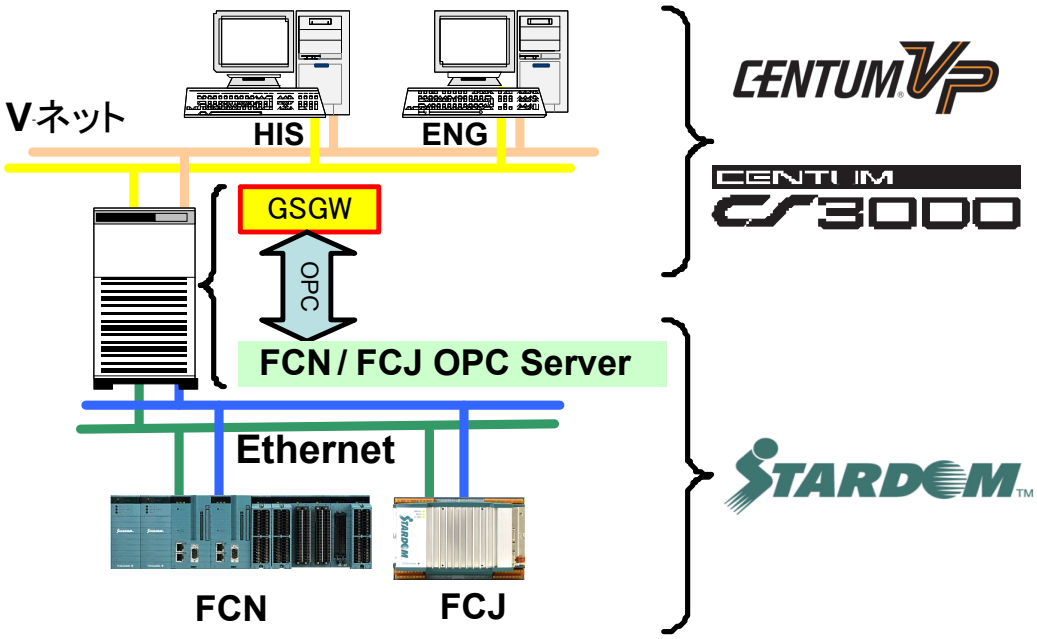


図　GSGW 接続例

表　汎用サブシステムゲートウェイ（GSGW）による CENTUM 接続（データ統合）

通信方式	システム	ステーション／接続ポート［参照ドキュメント］		オプションソフトウェア［参照ドキュメント］	備考
OPC 通信（Ethernet 通信）	CENTUM	GSGW をインストールした汎用 PC CENTUM VP: ［GS 33M20F10-40］ CS 3000: ［GS 33Q03P10-31］		LGS1250　GSGW 汎用サブシステムゲートウェイパッケージ CENTUM VP: ［GS 33M20F10-40］ CS 3000: ［GS 33Q03P10-31］	通信の二重化も可能
	FCN/FCJ	FCN	ネットワークポート（100BASE-TX）［GS 34P02Q12-01］	FCN/FCJ OPC サーバ for Windows［GS 34P02Q61-01］ FCN/FCJ OPC サーバ用ネットワーク二重化プログラム［GS 34P02Q62-01］	
		FCJ	ネットワークポート（100BASE-TX）［GS 34P02Q11-01］		

B2.2.4 CENTUM 接続（オペレーション統合）

UGS または SIOS を用いて FCN/FCJ を CENTUM システムに接続すれば、FCN/FCJ の機能ブロックデータを、HIS 上で 1 つのまとまりのあるデータとしてハンドリングできます。これにより、UGS 上または SIOS 上に写像された FCN/FCJ の機能ブロック（計器）を、CENTUM の機能ブロックと同様の Look &Feel で操作監視できます。
UGS を用いれば、FCN/FCJ のアラームを HIS 上で管理できます。
SIOS と統合型アラーム管理機能（CAMS for HIS）を組み合わせて用いれば、FCN/FCJ の接続システムが発信するアラームも一括して HIS 上で管理できます。

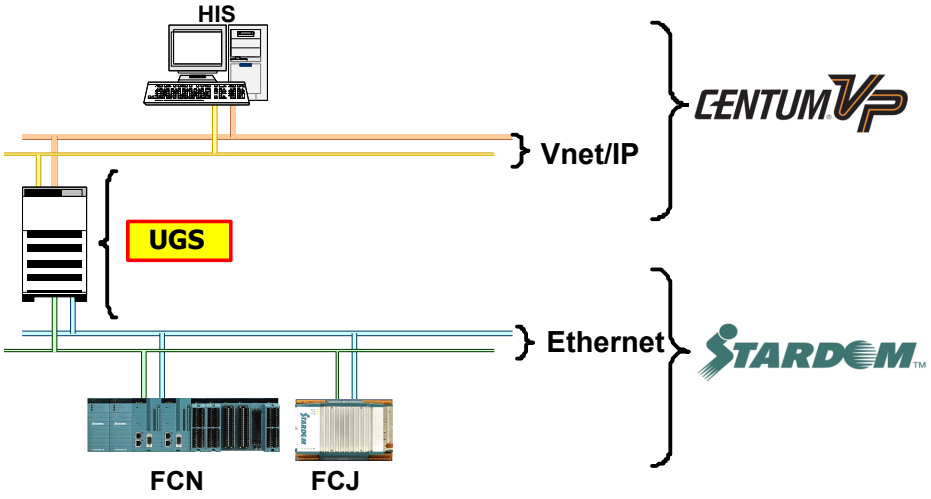


図　UGS を介して CENTUM に FCN/FCJ を接続した例

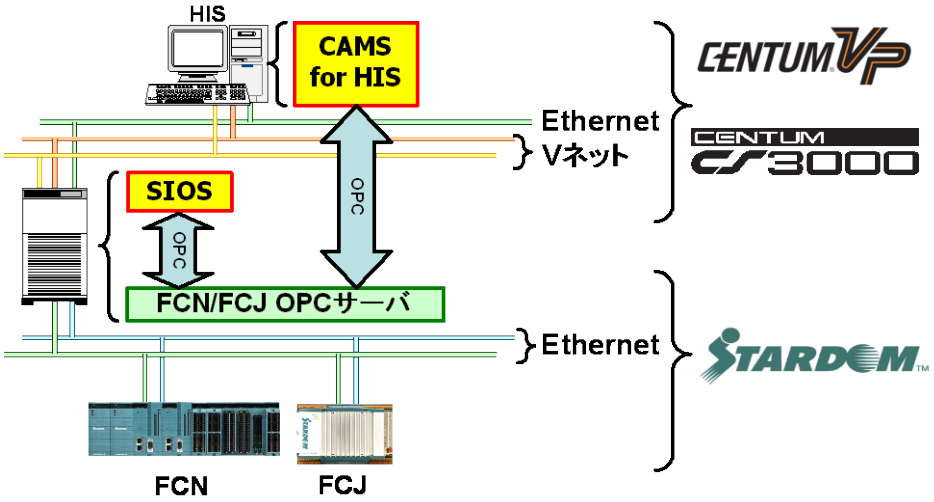


図　SIOS を介して CENTUM に FCN/FCJ を接続した例

補足

CENTUM VP では、CAMS for HIS は操作監視基本機能（LHS1100/LHM1101）に含まれます。
CENTUM CS 3000 では、オプションパッケージ（統合型アラーム管理パッケージ（LHS4800））が必要です。

表　UGS との接続方法

通信方式	システム	ステーション／接続ポート［参照ドキュメント］		ソフトウェア［参照ドキュメント］	備考
Ethernet 通信	CENTUM	UGS をインストールしたコンピュータ CENTUM VP: ［GS 33K20C10-50］		UGS 統合ゲートウェイステーション基本機能 CENTUM VP: ［GS 33K20C10-50］	通信の二重化も可能
	FCN/FCJ	FCN	ネットワークポート（100BASE-TX）［GS 34P02Q12-01］	-	
		FCJ	ネットワークポート（100BASE-TX）［GS 34P02Q11-01］		

表　SIOS／CAMS for HIS との接続方法

通信方式	システム	ステーション／接続ポート［参照ドキュメント］		ソフトウェア［参照ドキュメント］	備考
OPC 通信（Ethernet 通信）	CENTUM	SIOS をインストールしたコンピュータ CENTUM VP: ［GS 33M20D10-40］ CS 3000: ［GS 33Q05P10-31］		システム統合 OPC クライアントパッケージ CENTUM VP: ［GS 33M20D10-40］ CS 3000: ［GS 33Q05P10-31］	通信の二重化も可能
		CAMS for HIS をインストールした HIS CENTUM VP: ［GS 33M05D10-40］ CS 3000: ［GS 33Q02N03-31］		統合型アラーム管理パッケージ CAMS for HIS CENTUM VP: ［GS 33M05D10-40］ CS 3000: ［GS 33Q02N03-31］	
	FCN/FCJ	FCN	ネットワークポート（100BASE-TX）［GS 34P02Q12-01］	FCN/FCJ OPC サーバ for Windows ［GS 34P02Q61-01］ FCN/FCJ OPC サーバ用ネットワーク二重化プログラム ［GS 34P02Q62-01］	
		FCJ	ネットワークポート（100BASE-TX）［GS 34P02Q11-01］		

補足

- CENTUM VP R5.01 以降のオペレーション統合には、UGS の使用を推奨します。
- 本書の PART-B および PART-C では、CENTUM VP R4.01 の SIOS を例として、オペレーション統合について説明します。R4.01 とそれ以降では、SIOS エンジニアリングウィンドウのイメージが少し異なる場合があります。必要に応じて、製品に同梱の IM で確認してください。

参照

- UGS の詳細については、「統合ゲートウェイステーションリファレンス」（IM 33K03R30-50）を参照してください。
- SIOS の詳細については、「リファレンス（機能ブロック編/エンジニアリング編/HIS 編/システム統合 OPC ステーション編）」（IM 33M01A30-40）を参照してください。
- CENTUM 接続（オペレーション統合）の詳細については、「CENTUM/STARDOM 統合エンジニアリングガイド」（TI 34P02K41-01）を参照してください。

B2.2.5 他の FCN/FCJ、各種 PLC、Modbus 通信機器との接続

FCN/FCJ 間のデータを、パソコンを経由せずに、ダイレクトに連携できます。さらに、オプションパッケージにより、当社「FA-M3」、三菱電機「MELSEC」、オムロン「SYSMAC」といった PLC、Modbus 通信対応機器および DNP3 通信対応機器との通信も実現します。（「FA-M3」では、Ethernet 通信または RS-232-C シリアル通信が可能です。「MELSEC」では、Ethernet 通信のみが可能です。「SYSMAC」では、RS-232-C シリアル通信のみが可能です。）
FCN/FCJ は、最大 15 台の FCN/FCJ および最大 32 台の他（FCN/FCJ でない）コントローラと Ethernet 通信できます。

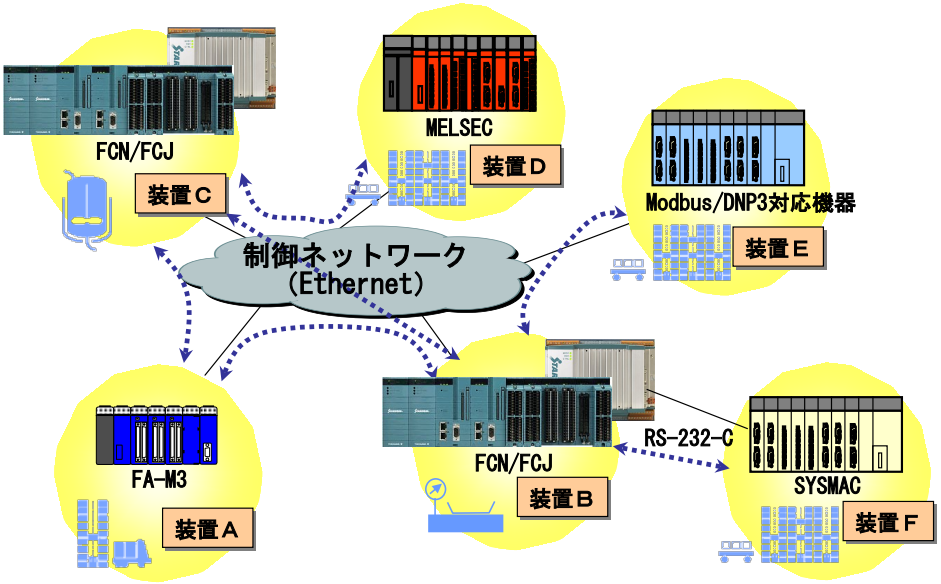


図　FCN/FCJ と各種コントローラ間通信

B2.2.6 表示器との接続

FCN/FCJ は、FA-M3 エミュレーション機能や Modbus 通信を用いて、表示器とも簡単に接続可能です。

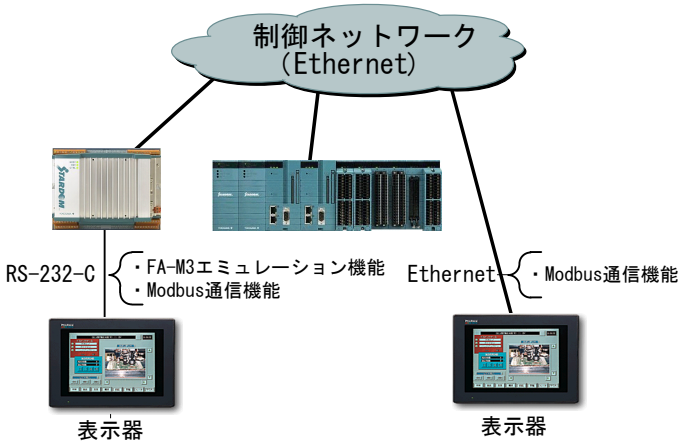


図　表示器との接続

B2.2.7 各種フィールド機器との接続

FCN/FCJ は、各種フィールド機器との接続のサポートによって、様々なアプリケーションに対応できます。また、フィールドレベルでのネットワーク分散を実現する FOUNDATION fieldbus もサポートしています。

● 現場でのオペレーションを実現する温調計/表示器接続

温調計通信ポートフォリオにより、横河電機製デジタル指示調節計 UT350／UT351／UT320／UT450／UT420／UT550／UT520／UT750 を容易に接続可能（最大 16 台／1 系統）

参照

温調計通信ポートフォリオについては、「B4.1.4 アプリケーションポートフォリオ」を参照してください。

表示器は、弊社「TOP3301S、TOP3600T」など FA-M3 エミュレーション機能／Modbus 通信により接続可能なものをサポートしています。

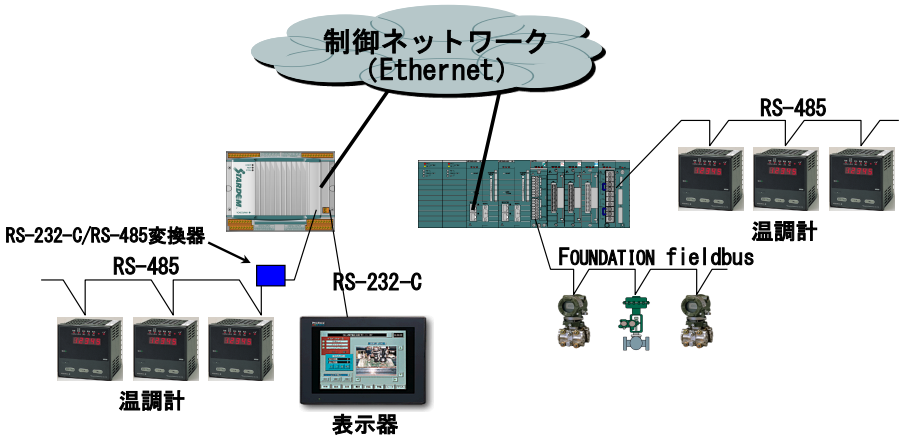


図　フィールド機器との接続

B2.3 FCN/FCJ の自律発信機能

**FCN/FCJ は、下表のような自律的に情報を発信する機能があります。
この自律発信機能の一部は、情報発信パッケージ「InfoWell」の利用により、プログラミングレスで、Web や E メール送信機能を実現できます。**

表 FCN/FCJ の自律発信機能

機能	説明	プログラム開発環境	
		InfoWell (*1)	Java AP (*2)
Web サーバ機能	FCN/FCJ が Web サーバとして、ネットワーク上の汎用 PC の Web ブラウザへデータを発信する機能	○	○
ロギング機能	FCN/FCJ が制御アプリケーションのデータをログファイルに保存する機能	○	○
E メール送信機能	FCN/FCJ が自らがアラームなど重要な情報を携帯端末などに E メール送信する機能	○	○
E メール受信機能	FCN/FCJ が受信した診断要求 E メールに応じて機器診断結果などを返信する機能	×	○
FTP 機能	FCN/FCJ が外部システム間とでデータファイルを受け渡しする機能	×	○
時刻同期機能（サーバ）	FCN/FCJ が、SNTP サーバとしてクライアントに時刻を配信する機能 (*3)	○	○
時刻同期機能（クライアント）	FCN/FCJ が、SNTP クライアントとして SNTP サーバと時刻を同期する機能 (*4)	○	○

*1: 情報発信パッケージ「InfoWell」で実現できる機能です。
*2: Java アプリケーションの作成により実現できる機能です。
*3: 時刻同期サーバポートフォリオのライセンスが必要です。
*4: FCN/FCJ 標準の機能です。

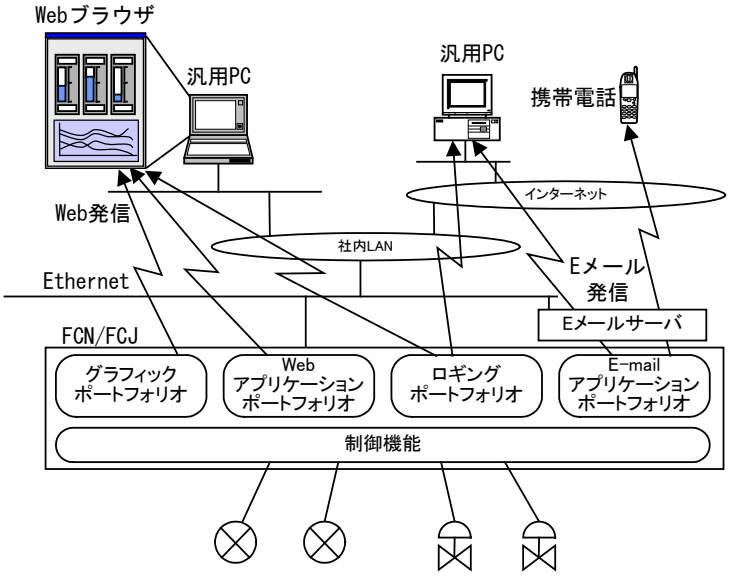
参照

- 情報発信パッケージ InfoWell については、「B2.3.1 情報発信パッケージ InfoWell」を参照してください。
- Java アプリケーションを作成については、「B2.3.2 Java アプリケーション」を参照してください。
- 時刻同期機能については、「B2.3.3 時刻同期機能」を参照してください。

B2.3.1 情報発信パッケージ InfoWell

情報発信パッケージ（InfoWell）は、FCN/FCJ 上で動作するパッケージで、SCADA ソフトを必要とせずに FCN/FCJ 単体で、Web や E メールによる情報発信の機能を提供します。
本パッケージを使用すると SCADA ソフトウェアによる常時監視を必要としないような設備でも FCN/FCJ 単体での設備監視・操作ができるようになります。
InfoWell は、以下のポートフォリオがあります。

- グラフィックポートフォリオ
  FCN/FCJ を Web サーバとして、ネットワーク上の汎用 PC の Web ブラウザにグラフィカルな画面とデータを表示するポートフォリオです。
- Web アプリケーションポートフォリオ
  FCN/FCJ を Web サーバとして、ネットワーク上の汎用 PC の Web ブラウザへデータを発信するポートフォリオです。
- E-mail アプリケーションポートフォリオ(*1)
  FCN/FCJ にメッセージ、または警報が発生したときに FCN/FCJ から自動的に E メールを発信するポートフォリオです。
- ロギングポートフォリオ
  FCN/FCJ 上で制御アプリケーションのデータをログファイルに保存して、FCN/FCJ を Web サーバとして、ネットワーク上の汎用 PC の Web ブラウザにログファイルの内容を表示するポートフォリオです。



*1: E メール発信のためには、E メールサーバが別途必要です。
イントラネット内に設置された E メールサーバだけでなく、インターネットプロバイダなどの外部の E メールサーバを利用することも可能です。外部の E メールサーバを利用する場合は、イントラネットのファイアウォールを外部 E メールサーバに E メール送信できるように設定してください。

図 InfoWell の機能概要図（イントラネット内の E メールサーバを使用したときの例）

● 特長

- 本パッケージの使用により、FCN/FCJ に情報発信機能を追加できます。
- グラフィックポートフォリオを使用すると、設備の写真やループ図などの画像中に FCN/FCJ の制御／監視対象データを数値やバー表示などの形で表示できるので、直感的に設備を監視できるようになります。
- グラフィックポートフォリオや Web アプリケーションポートフォリオを使用すると、FCN/FCJ を Web サーバとして、ネットワーク上のすべての汎用 PC から設備の監視・操作ができるようになります。
- E-mail アプリケーションポートフォリオを使用すると、OA エリアの汎用 PC や携帯電話へ E メールを発信できるので、いつでも、どこでも設備からのメッセージ、データを受け取れるようになります。
- グラフィックポートフォリオや Web アプリケーションポートフォリオと E-mail アプリケーションポートフォリオを組み合わせると、通常は FCN/FCJ での設備の無人運転、設備異常により E メール発信、E メールを受信した汎用 PC にて Web ブラウザによる設備の異常状態確認／操作といった対応が可能になります。
- Web アプリケーションポートフォリオと E-mail アプリケーションポートフォリオの属性は、Web ブラウザから簡単に設定できます。プログラミングなど特別な知識を必要としないので、誰でも簡単に設定できます。
- ロギングポートフォリオを使用すると、設備や機器の制御システムの管理に不可欠な高信頼のロギング機能を汎用 PC 無しで安価に実現できるようになります。

参照

情報発信パッケージ InfoWell については、本書「F3. GS 集：APPF 関連」「情報発信パッケージ（InfoWell）」（GS 34P02P51-01）、「グラフィックポートフォリオ（InfoWell）」（GS 34P02P52-01）、「STARDOM ロギングポートフォリオ（InfoWell）」（GS 34P02P53-01）を参照してください。

● グラフィックポートフォリオと Web アプリケーションポートフォリオの違い

グラフィックポートフォリオと Web アプリケーションポートフォリオの違いがあります。

表　グラフィックポートフォリオと Web アプリケーションポートフォリオの違い

機能	グラフィック ポートフォリオ	Web アプリケーション ポートフォリオ
画面	グラフィカルな画面を簡単に作成し、発信が可能	データの参照／設定が可能 トレンド、エラー履歴が表示可能
開発環境	汎用 PC 上（専用ツール）	FCN/FCJ（Web ブラウザ経由）

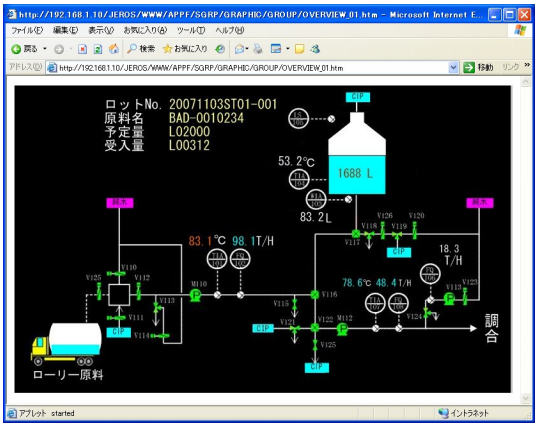
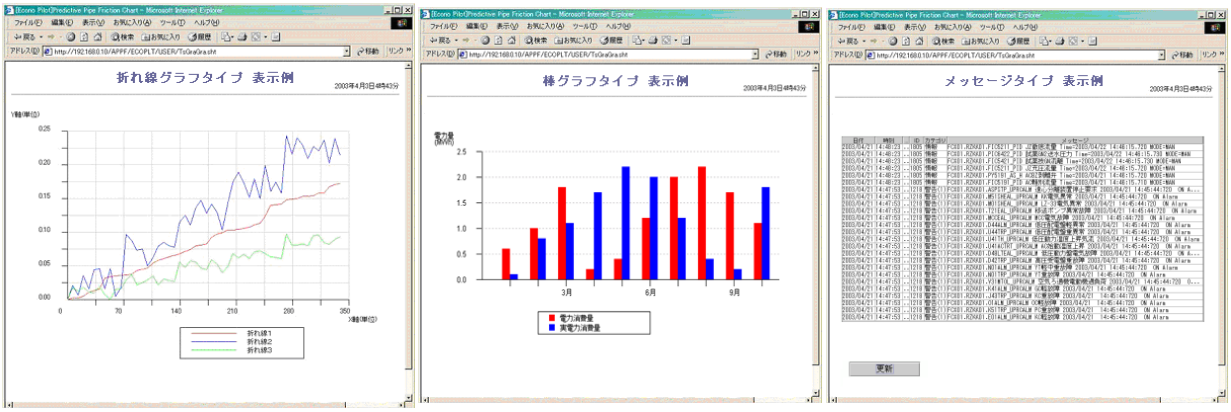


図　InfoWell（グラフィックポートフォリオ）の画面表示例



折れ線グラフ　　棒グラフ　　メッセージ表示

図　InfoWell（Web アプリケーションポートフォリオ）の画面表示例

■ グラフィックポートフォリオ

グラフィックポートフォリオ（InfoWell）は、Web ブラウザでグラフィック画面を表示できます。
グラフィック画面には、ビットマップ形式（BMP）、JPEG 形式、GIF 形式、PNG 形式などの画像ファイルを背景画として表示し、この背景画の上に表示部品を配置できます。この表示部品は、FCN/FCJ のデータ（制御プログラムの変数値）により表示内容を変更できます。

● 表示部品

グラフィック画面に表示可能な部品を下記に示します。

表　表示部品一覧

表示部品	主な機能
データ表示部品	データを数値で表示
テキスト表示部品	データにより選択されたテキストを表示
イメージ表示部品	データによりイメージを変更
バー表示部品	データによりバーを増減
ラベル／リンク部品	ラベル表示、リンクを表示

・ データ表示部品

データを数値で表示します。さらにデータ値により背景色を変更できます。

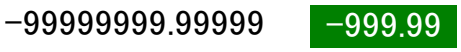


図 データ表示部品の例

・ テキスト表示部品

データ（0～7 または True／False）により指定したテキストを表示します。さらにデータによりフォント色または背景色を変更できます。

図 テキスト表示部品の例

・ イメージ表示部品

データ（0～7、または True／False）によりイメージを切り替えて表示します。さらに、イメージの回転や反転もできます。

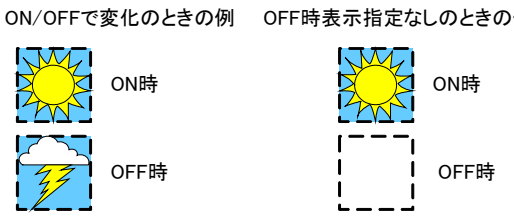


図 イメージ表示部品の表示例

・ **バー表示部品**

データをバー形式で表示します。データによりバーを増減させて表示するほか、色を変化させて表示できます。

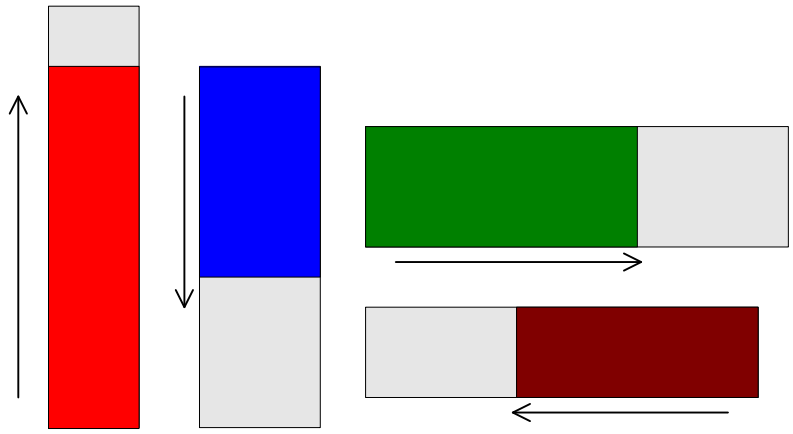


図 バー表示部品の例

・ **ラベル／リンク部品**

ラベル／リンク部品は、ラベルや、他のグラフィックや URL に展開するボタンを表示します。

図 ラベル／リンク部品の例

● データ表示機能

データ表示の手順は、以下のとおりです。

1. 汎用 PC の Web ブラウザから FCN/FCJ の URL を指定します。
2. Web ブラウザは、HTML に指定された画面情報（背景画や表示部品など）を FCN/FCJ からアップロードします。
3. Web ブラウザ上に、背景画を表示します。
4. Web ブラウザ上の表示部品が FCN/FCJ にデータを要求します。
5. FCN/FCJ のデータ転送機能が制御アプリケーションのデータを読み出し、Web ブラウザに送信します。
6. Web ブラウザは返信されたデータにより、表示部品の情報を更新します。

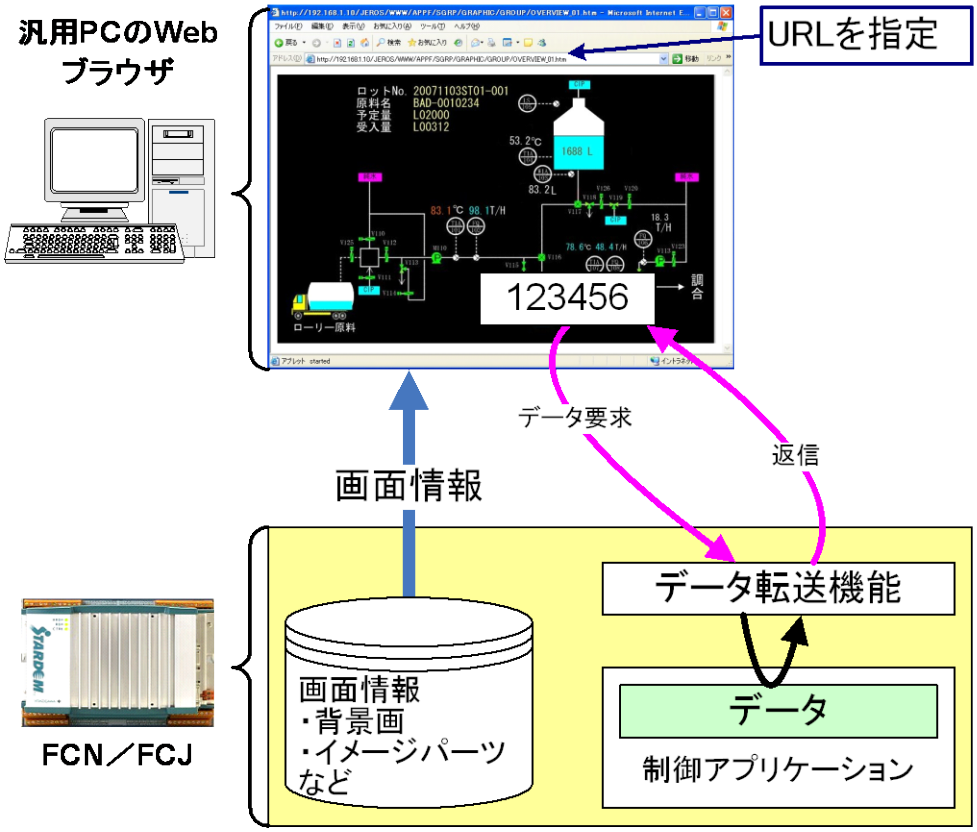


図　データ表示機能の動作イメージ

補足

- データを表示する部品は、データ表示部品、テキスト表示部品、イメージ表示部品、バー表示部品です。ラベル／リンク部品は、データと連動しません。
- 1 つの画面に複数（最大 10 台）の FCN/FCJ のデータを表示可能です。
- 1 台の FCN/FCJ は、最大 10 画面を表示できます。

■ Web アプリケーションポートフォリオ

Web アプリケーションポートフォリオでは、Web 画面表示を以下の 3 つの実現方式により実現し、各々特長を持った画面を提供しています。

表　実現方式一覧

実現方式	説明
テーブル表示方式	データをテーブル（表）にて表示 データの表示と設定が可能
グラフィカル表示方式	データをグラフなどグラフィカルに表示 データの定期更新が可能
リアルタイム更新方式	データをリアルタイムに更新して表示

表　画面一覧

実現方式	画面	説明
テーブル表示方式	テーブル	複数のデータを 1 つの表にて表示
グラフィカル表示方式	棒グラフ	配列データを棒グラフにて表示
	折れ線グラフ	配列データを折れ線グラフにて表示
	メッセージ表示	FCN/FCJ のシステムメッセージ・アプリケーションメッセージを時系列に表示
リアルタイム更新方式	数値表示	データを数値によりリアルタイム表示
	バー表示	データをバーによりリアルタイム表示
	トレンド表示	データの時系列変化をトレンドにて表示

本ポートフォリオでは、これらの画面を必要に応じて選択して使用します。

● テーブル表示方式

本方式では、HTML によるテーブル（表）にて表示します。
本方式では、データの表示・設定が可能です。

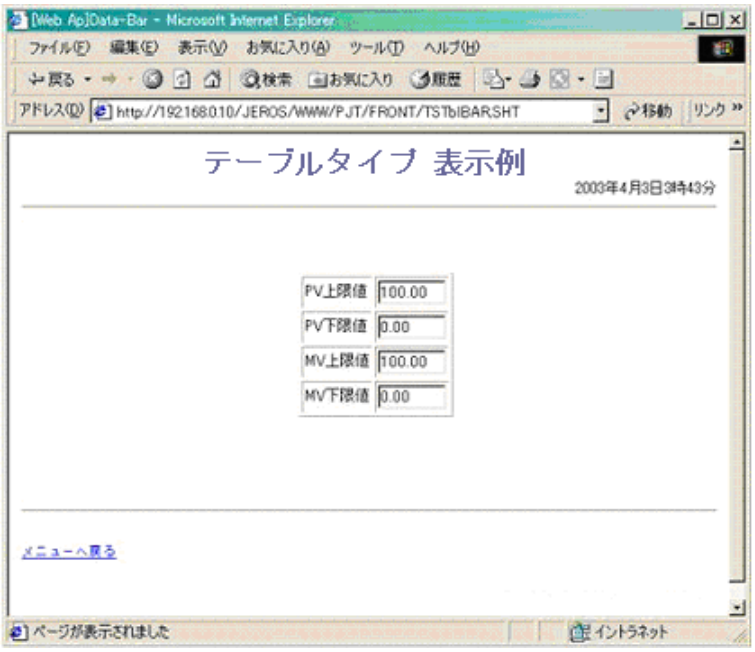
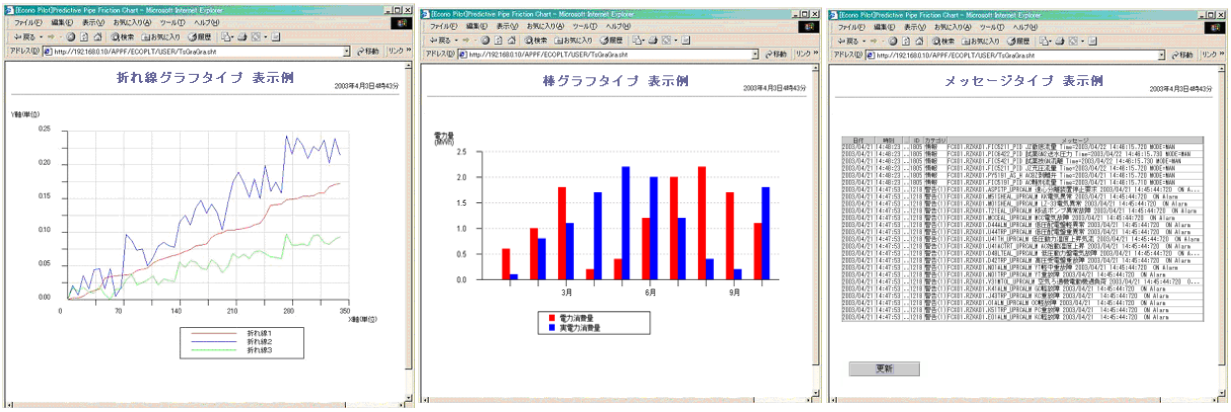


図　テーブル表示方式の画面イメージ

● グラフィカル表示方式

本方式では、Web ブラウザ上で実行されるプログラム（Java アプレット）によりデータを表示します。
本方式では、データを定期的に更新できます。（メッセージ表示は定期更新不可）
本方式のデータ更新方法は PULL 型の定期更新で、Java アプレットが定期的に FCN/FCJ へデータを要求します。

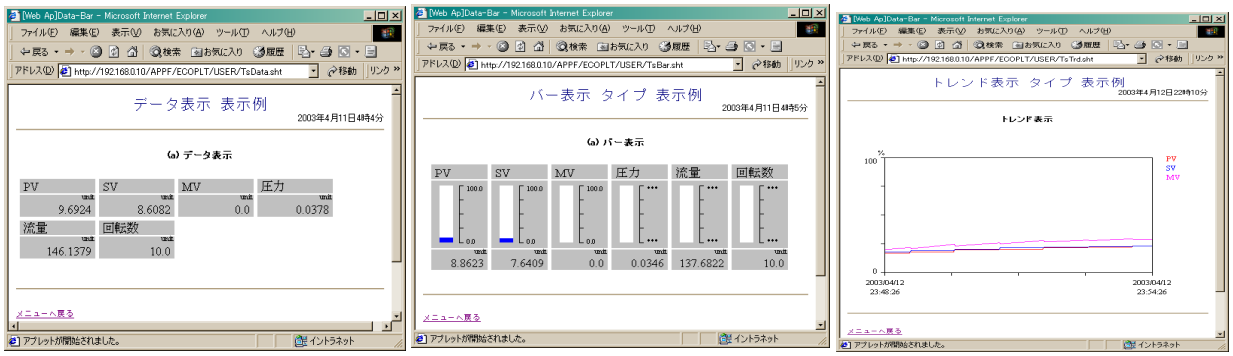


折れ線グラフ　　棒グラフ　　メッセージ表示

図　グラフィカル方式の画面イメージ

● リアルタイム更新方式

リアルタイム更新方式は、Web ブラウザ上で実行されるプログラム（Java アプレット）によりデータを表示します。
本方式では、データをリアルタイムに更新できます。
本方式のデータ更新方法は PUSH 型の定期更新で、FCN/FCJ 上のデータ収集・蓄積用プログラムがデータを定期収集し Web ブラウザ上の Java アプレットに定期送信します。



数値表示　　バー表示　　トレンド表示

図　リアルタイム更新方式の画面イメージ

■ E-mail アプリケーションポートフォリオ

E-mail アプリケーションポートフォリオでは、FCN/FCJ のイベント発生時に自動的に E メールを発信する機能を提供します。
E-mail アプリケーションポートフォリオでは、以下の 2 種類の E メール発信が用意されています。

● 警報発信

警報発信は、制御アプリケーションのデータの上下限値オーバと接点状態を検出し、E メールを発信します。ここでの警報検出は、制御アプリケーションによる警報検出とは別で、E-mail アプリケーションポートフォリオ側における独自のデータ収集・警報検出です。
発信する E メールの本文は、制御アプリケーションのデータごとに個別に設定可能です。そのため、警報検出をトリガとして自由な文章の E メールを発信できます。

```
FCX1 192.168.1.1
2003/02/08 0:00:00
タンク内温度がHIGH_ALARM状態です。現在値は120℃です。
スイッチ１がBOOL_ALARM状態です。現在値はtrueです。
        ・
        ・
        ・
```

図　警報発信 E メール本文例

● メッセージ発信

メッセージ発信は、FCN/FCJ のシステムメッセージ、アプリケーションメッセージ（制御アプリケーションからのメッセージ）の発生時に E メールを発信します。

```
FCX1 192.168.1.1
2003/02/08 0:00:00
Type = application alarm Occur, Application alarm recover
Instance Name = Task1.Loop1
Key Word = HH,LL
2003/02/27 20:19:10 alarm
message.application.Task1.Loop1.2206.HH alarm(PV=88%)
        ・
        ・
        ・
```

図　メッセージ発信 E メール本文例

■ ロギングポートフォリオ

ロギングポートフォリオは、設備や機器の制御システムの管理に不可欠なロギング機能を実現するもので、自律型コントローラ FCN/FCJ の制御アプリケーションのデータを、システムの用途に最適な各種の形式で収集し、ロギングファイル（CSV 形式）として保存します。このロギングファイルは、ネットワーク経由で Web ブラウザから各種の形式（トレンド表示、表形式表示など）で閲覧できます。さらに、E メールで送信できます。同梱されている、汎用 PC 用ソフトウェアを活用すれば、このロギングファイルを、汎用 PC からネットワーク経由で定期取得して活用できます。

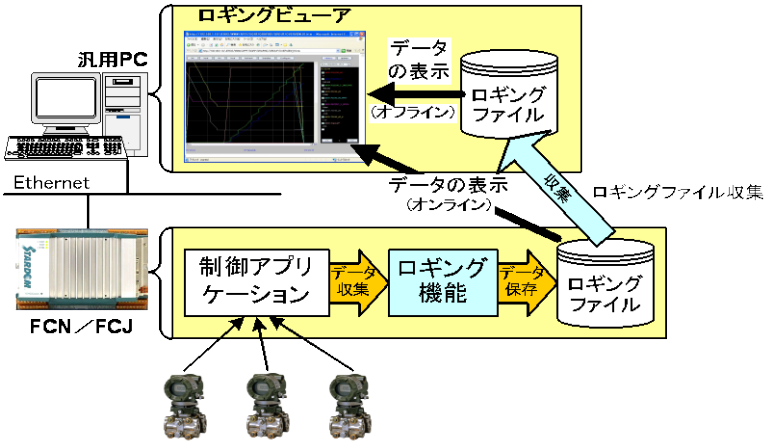


図　ロギング機能

● ロギング機能

- データロギング

表　データロギング

ロギングタイプ		データ収集周期		締め切りタイミング	簡易レポート(*4)
		高速	低速		
時締めロギング		1～60 秒 (*1)(*3)	1～60 分 (*2)	1 時間ごと	日報、月報、年報
日締めロギング		―	1～60 分 (*2)	1 日ごと	月報、年報
バッチロギング		1～60 秒 (*1)(*3)	1～60 分 (*2)	バッチ記録スイッチ	―
スナップショットロギング	連続	―	トリガ	1 時間または 1 日ごと	日報、月報、年報
	トリガ			締め切りトリガ	―

*1:　既定値（1、2、5、10、15、30、60 秒）が指定可能
*2:　既定値（1、2、5、10、15、30、60 分）が指定可能
*3:　データ収集には、制御アプリケーション作成時にロギング用 POU が必要です。
*4:　日報、月報、年報を含む簡易レポートは、締め切り値（最大、最小、平均、合計）から成る CSV 形式ファイルです。

- シーケンスイベントロギング（SOE ロギング）
  異常などのイベントが発生した前後のデータを高速にロギングし、ファイルへ保存します。簡易レポートは作成できません。
- メッセージロギング
  FCN/FCJ のメッセージを定周期（1 分）にファイルに保存します。メッセージ番号によりフィルタリングもできます。

● ロギングファイル収集機能

ロギングファイル収集機能を用いて、定期的に FCN/FCJ のロギングファイルを汎用 PC に収集できます。また、手動による任意タイミングでの収集も可能です。
ロギングファイルの収集機能には、次の方法があります。
用途に応じて、選択して使用してください。

- ロギングファイル収集ツール：ログインしてタスクトレイに常駐した状態で動作します。
- ロギングファイル収集サービス：ログインなしでバックグラウンドで動作します。

●ロギングファイル表示機能

- ロギングファイル表示機能
  ロギング機能が収集したデータをロギングビューアにより、Web ブラウザ上で表示できます。（オンライン表示）
  FCN/FCJ 内のロギングファイルだけでなく、汎用 PC 上に収集されたロギングファイルも、ロギングビューアを用いて簡単に表示できます。（オフライン表示）
  ロギングビューアでの表示方法には、トレンド表示と表形式表示があります。
- メッセージ表示機能
  収集した FCN/FCJ のメッセージロギングもデータと同様にロギングビューアで表示できます。

● ロギングファイル送信（E メール）機能

ロギングファイル送信機能を用いて、ロギングファイルを E メールの添付ファイルとして送信できます。

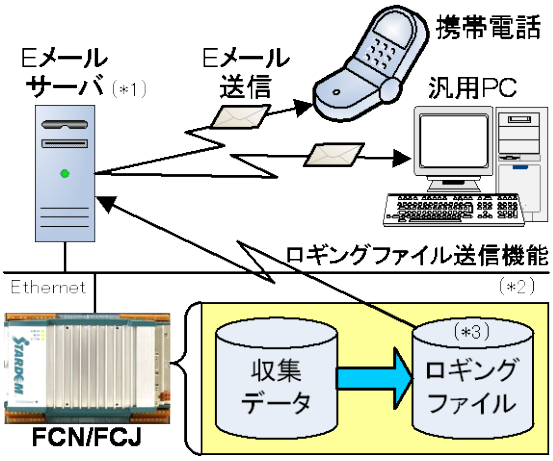


図　ロギングファイル送信（E メール）機能概念図（イントラネット内の E メールサーバを使用したときの例）

*1: E メール発信のためには、E メールサーバが別途必要です。
イントラネット内に設置された E メールサーバだけでなく、インターネットプロバイダなどの外部の E メールサーバを利用することも可能です。外部の E メールサーバを利用する場合は、イントラネットのファイアウォールを外部 E メールサーバに E メール送信できるように設定してください。

*2: FCN/FCJ のメッセージや警報を E メールで発信する場合には、「E-mail アプリケーションポートフォリオ（InfoWell）」を利用してください。
「ロギングファイル送信（E メール）機能」と「E-mail アプリケーションポートフォリオ（InfoWell）」は、併用できます。

*3: 送信可能なロギングファイルは、簡易レポート（日報、月報、年報）、バッチロギング、スナップショットロギング、SOE ロギングです。

B2.3.2 Java アプリケーション

Java アプリケーションの作成により、下記の任意の機能を作成できます。

- **Web サーバ機能**
- **E メール送受信機能**
- **FTP 機能**

重 要

Java アプリケーションを作成すれば、任意の情報発信機能を構築できます。
ただし、情報発信パッケージ「InfoWell」を利用すれば、プログラミングレスで簡単に情報発信機能を構築できます。

参照

Java アプリケーションの開発環境については、「B4.1.5 FCN/FCJ Java アプリケーション開発」を参照してください。

■ Web サーバ機能

FCN/FCJ を WWW サーバとして、汎用パソコンの Web ブラウザにて FCN/FCJ の収集データなどを監視できます。

- FCN/FCJ 上に作成した HTML ファイルや Java アプレットは、アクセスしてきた Web ブラウザへダウンロードされ収集データなどの監視が可能となります。クライアントとなる Web ブラウザが動作する PC には、特別なソフトウェアは必要ありません。
- イントラネットに限らずインターネット経由でもアクセス可能です。（別途 Firewall などの設定が必要です）
- 権限のない者の侵入を防ぐセキュリティを確保できます。（ユーザアカウント定義ファイルの設定が可能）

補足

- FCN/FCJ 上の HTML ファイルを保存したフォルダ単位でアクセス制御が可能
- ユーザ名とパスワードの入力によって HTML ファイルにアクセス可能

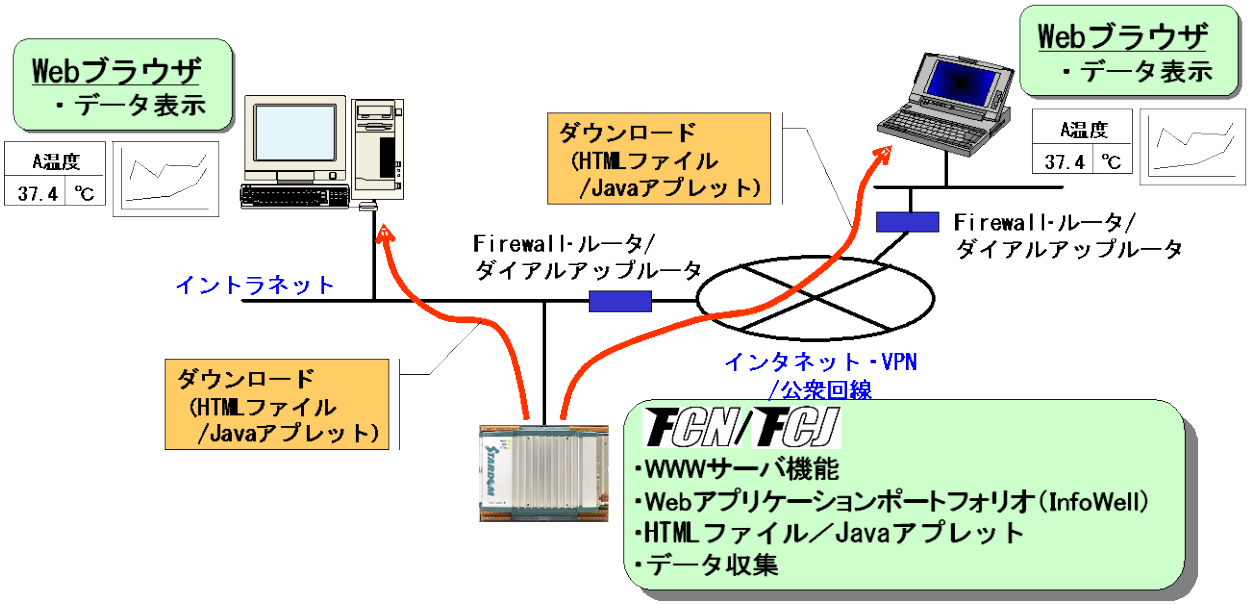


図　FCN/FCJ の WWW サーバ機能

■ E メール送受信機能

FCN/FCJ は、自らのアラームなど重要な情報を携帯端末に E メールで送信でき、受信した診断要求 E メールに応じて機器診断結果を返信できます。これにより、オペレータはどこにいても現場の状況を認識でき、操業効率の向上を図れます。

- 警報発生時に HMI ソフトウェアや人を介さずにオペレータへ E メール発信できます。
- オペレータからの E メールを FCN/FCJ が受信し、その E メールに対応した処理を実行できます。
- E メールサーバがインターネット E メールサーバの場合は、パソコンのほか携帯電話などでも E メールを受け取れます。
- E-mail アプリケーションポートフォリオ（InfoWell）を利用すれば、プログラミングは不要です。Web ブラウザからの簡単な設定だけで、簡単にデータ表示が可能になります。

補足

- E メールの送信は SMTP プロトコル、受信は POP3 プロトコルを使用します。
- 機器診断ロジックや対象機器状態収集ロジックなどアプリケーションを別途作成する必要があります。

参照

E-mail アプリケーションポートフォリオ（InfoWell）については、「B2.3.1 情報発信パッケージ InfoWell」を参照してください。

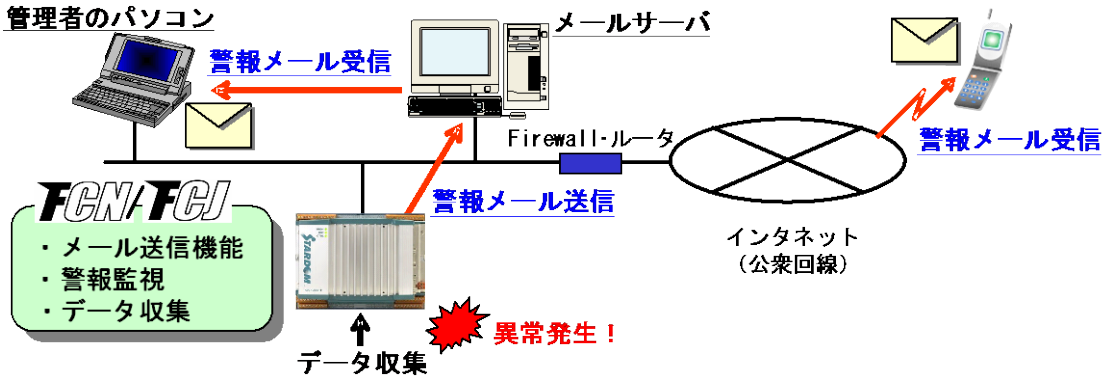


図　FCN/FCJ の E メール送信機能（例）

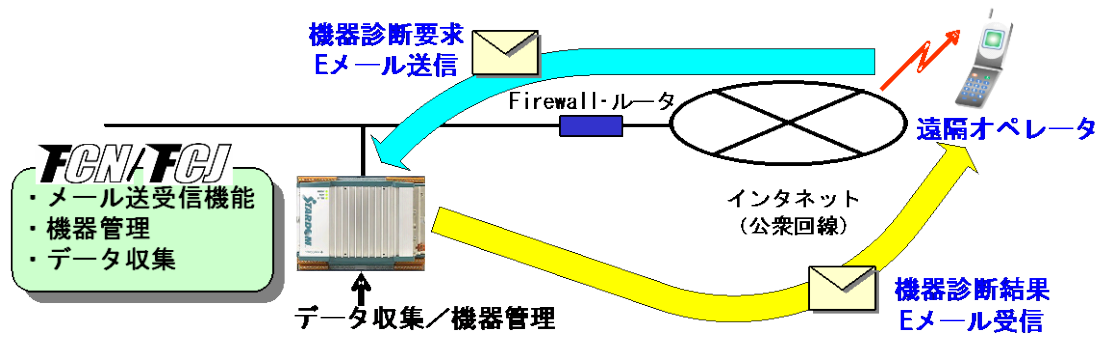


図　FCN/FCJ の E メール送受信機能（例）

■ FTP 機能

FCN/FCJ と外部システム間でデータファイルを受け渡せます。

- FTP クライアント機能を使用して FCN/FCJ から任意の FTP サーバへ、データファイルを転送できます。定期的に FCN/FCJ 上の収集データファイルを FTP サーバへ転送でき、警報が発生した際に FCN/FCJ 上の主要データファイルを FTP サーバへ転送できます。
- FTP サーバ機能により、外部よりデータファイルを取得できます。

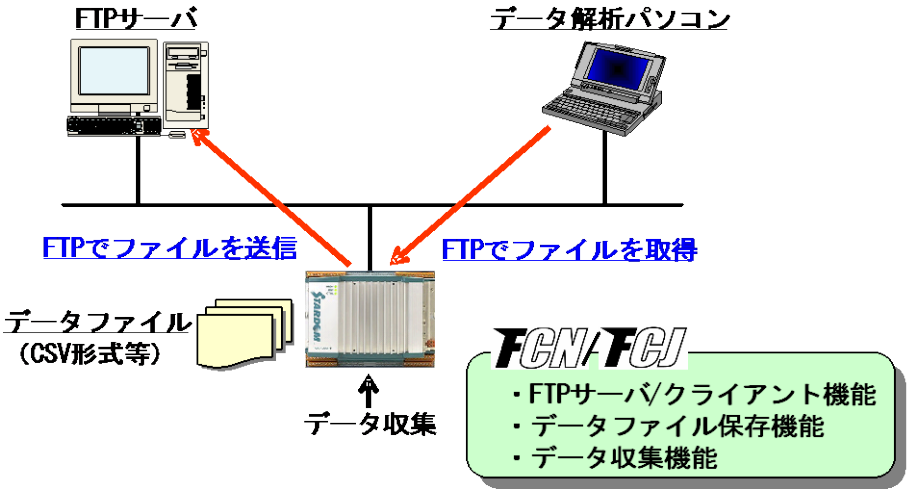


図　FCN/FCJ の FTP 通信機能

B2.3.3 時刻同期機能

FCN/FCJ は、SNTP（Simple Network Time Protocol）をサポートした機器間で時刻を同期できます。FCN/FCJ は、SNTP のサーバおよびクライアントとして動作可能です。

■ 機能仕様

表　時刻同期機能

項目	機能	
	サーバ機能	クライアント機能
通信プロトコル	SNTP（Simple Network Time Protocol）(*1) UDP ポート：123	
ライセンス	必要(*2)	不要
ユニキャストモード(*3)	○	○
ブロードキャストモード(*4)	－	○
接続数	32 接続以下を推奨(*5)	4 サーバ
時刻精度	FCN/FCJ の内部タイマの精度に依存(*6)	±500 ms (*8)
時刻補正方法	－	100 秒周期でサーバから時刻を取得（ユニキャストモード）(*7)
その他	・ネットワークの二重化時も対応可能 ・CPU の二重化時も対応可能	

*1:　サーバ機能とクライアント機能の同時実行可能。
*2:　時刻同期サーバポートフォリオライセンスが必要です。
*3:　SNTP クライアントが定周期で SNTP サーバに時刻を問い合わせます。
*4:　SNTP サーバが定周期に SNTP クライアントに時刻を通知します。
*5:　クライアントの接続数に制限はありませんが、FCN/FCJ の負荷を考慮して接続してください。
*6:　FCN/FCJ の内部タイマの誤差（-17.5 ～ + 12 s／日）を含む時刻が、SNTP クライアントに時刻として通知されます。連携する機器の時刻を同期するものであり、高精細度の時刻は提供できません。
*7:　時刻差分に応じ、以下の様に時間を調整します。（下記時刻差分は、初期値です。時刻差分はコンフィギュレーション可能です。）
500 ms 以下：　調整しません
5 s 未満：　スムーズ調整（時刻が飛ばないようにスムーズに時間を調整します）
5 s 以上：　一気設定（一気に指定時刻を設定します）
*8:　時刻補正方法のコンフィギュレーションにより、時刻の精度は変わります。

参照

時刻同期サーバポートフォリオについては、本書「F3. GS 集：APPF 関連」「FCN/FCJ アプリケーションポートフォリオ」（GS 34P02P20-01）を参照してください。

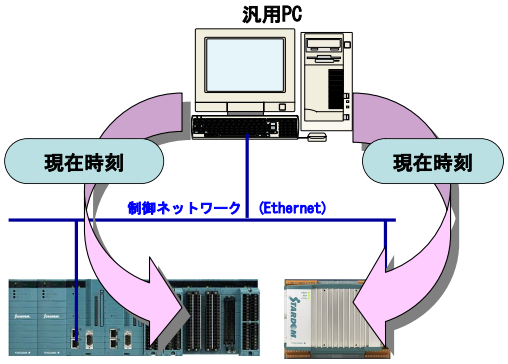


図　時刻同期機能（ブロードキャストモード）

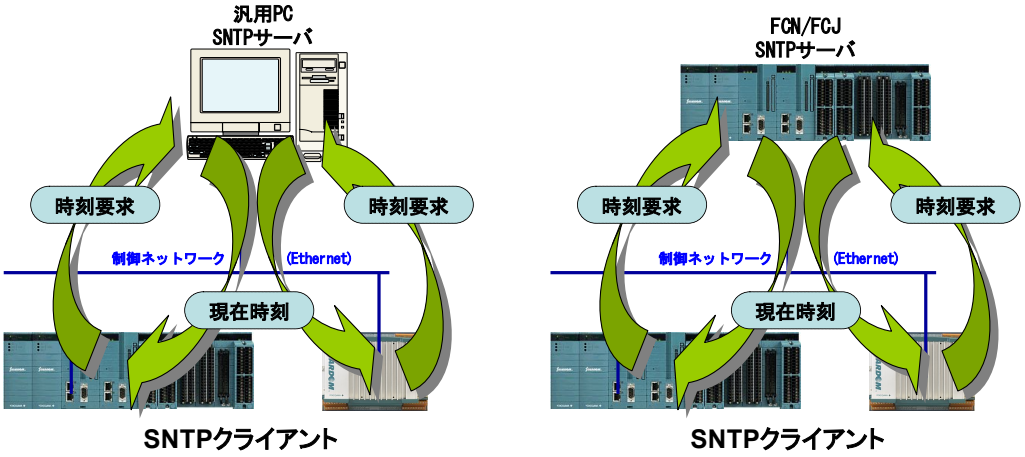


図　時刻同期機能（ユニキャストモード）

● 時刻同期機能（ブロードキャストモード）

時刻同期機能（ブロードキャストモード）では、SNTP サーバが同一のサブネット上のすべてのユニットに対して、一定周期ごとに現在時刻を通知します。SNTP クライアントがこの現在時刻を受信し、時刻を同期します。

補足

ブロードキャストモードには、以下の制約があります。
- 異なるサブネットのユニットとは、時刻同期できません。
- ラウンドトリップタイム（サーバ／クライアント間の通信に要する時間）を考慮できないため、遅い通信回線を使用した場合や通信負荷が高い場合には、時刻制度が悪くなるときがあります。
- 同一サブネット上に複数台のサーバを設置できません。

● 時刻同期機能（ユニキャストモード）

時刻同期機能（ユニキャストモード）では、SNTP クライアントが SNTP サーバに対して、時刻の問い合わせを要求します。これに対して、SNTP サーバは、SNTP クライアントに現在時刻を通知します。SNTP クライアントがこの現在時刻を受信し、時刻を同期します。

補足

ユニキャストモードには、ブロードキャストモードの持つ制約条件がありません。このため、ユニキャストモードでは、より精度の高い時刻同期機能を提供できます。FCN/FCJ の時刻同期には、ユニキャストモードの使用を推奨します。

■ 時刻補正方法

FCN/FCJ の時刻補正方法は、時刻差分により下記の 3 パターンがあります。

- 時刻を調整しない（時刻差分が 500ms 以下(*1)）
- スムーズ調整（時刻差分が 500ms 以上、5s 未満(*1)）
- 一気設定（時刻差分が 5s 以上(*1)）

*1: 上記、時刻差分に関するパラメータは、初期値です。時刻差分に関するパラメータは、コンフィギュレーション可能です。

● スムーズ調整

スムーズ調整は、定周期タイマの割り込み間隔を 0.5%調整し、時刻を補正します。SNTP クライアントの時刻が遅れている場合、定周期タイマの割り込み間隔を 0.5%早めて、時刻を進めます。SNTP クライアントの時刻が進んでいる場合、定周期タイマの割り込み間隔を 0.5%遅くし、時刻を遅らせます。時刻差分に関するパラメータが初期値の場合、スムーズ調整では、100 秒間に 500 ミリ秒、時刻を補正できます。

● 一気設定

SNTP サーバと SNTP クライアントの時刻に 5 s 以上の差があると、SNTP クライアントの時刻を、SNTP サーバの指定時刻にします。これにより、SNTP クライアントの時刻は連続性が無くなります。

B2.4 制御ネットワーク

FCN/FCJ では、制御ネットワークの「二重化」や「2 系統化」が可能です。

- 制御ネットワークの二重化
  制御ネットワークの二重化は、ネットワークの高信頼化のために使用します。
- 制御ネットワークの 2 系統化
  制御ネットワークの 2 系統化は、片側を制御ネットワークに接続し、もう一方のネットワークには、情報系 LAN やフィールドネットワークに接続して使用します。

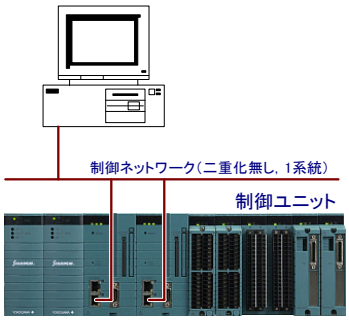


図　制御ネットワーク（二重化無し、1 系統）

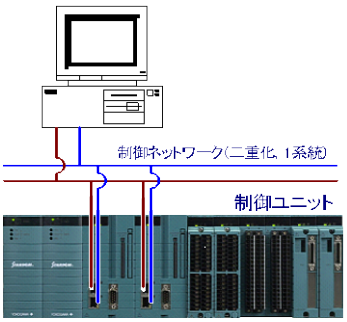


図　制御ネットワーク（二重化、1 系統）

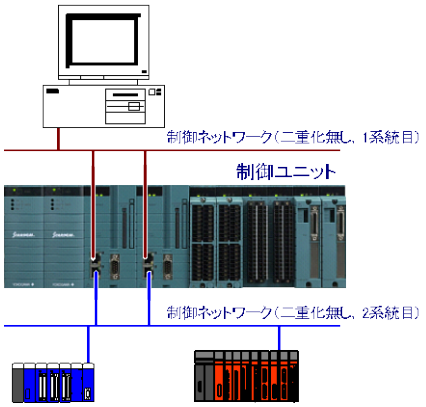


図　制御ネットワーク（二重化無し、2 系統）

B2.4.1 制御ネットワークの二重化

自律型コントローラ FCN/FCJ と VDS/ASTMAC が接続される制御ネットワークは、リアルタイム性と高い信頼性が要求されるため、ネットワーク二重化の機能をサポートしています。各ノード間における通信経路の二重化によって、万一通信経路に異常が発生した場合にも、瞬時にかつ自動的に通信経路を切り替えて通信を継続し、システムへの影響を最小限にできます。

参照

制御ネットワークの二重化についての詳細は、「B7.3 ネットワークの二重化」を参照してください。

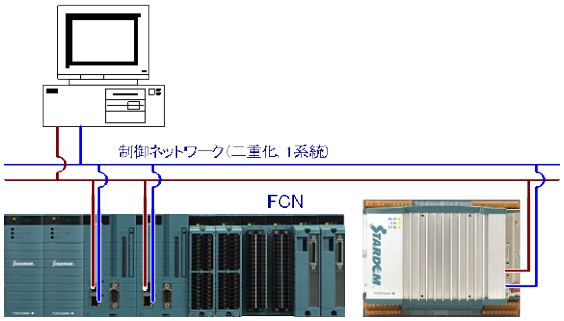


図　FCN/FCJ および VDS/ASTMAC 制御ネットワーク二重化の例

B2.4.2 制御ネットワークの 2 系統化

制御ネットワークの 2 系統化 FCN/FCJ には、2 つの Ethernet ポートを実装しています。この Ethernet ポートをそれぞれ別の制御ネットワークに接続できます。これにより、それぞれの制御ネットワークのトラフィックを減少させられます。

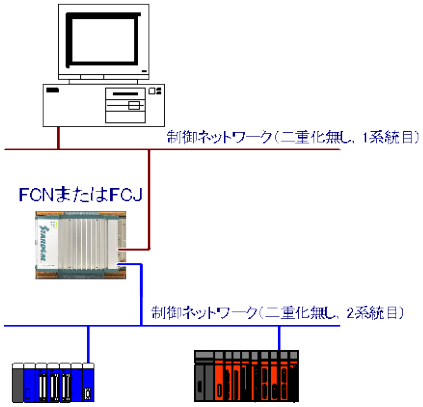


図　FCN/FCJ の制御ネットワークを 2 系統化して使用した例

B3. FCN/FCJ ハードウェア

本章では、自律型コントローラ FCN/FCJ のハードウェアについて説明します。

参照

FCN/FCJ の設置については、「PART-E FCN/FCJ 設置仕様編」を参照してください。

B3.1 FCN モジュール構成

FCN は以下のモジュールで構成されています。

- ベースモジュール（各モジュールを実装するバックボード）
- 電源モジュール
- CPU モジュール
- SB バスリピートモジュール（増設ユニット接続インタフェース）
- 各種 I/O モジュール

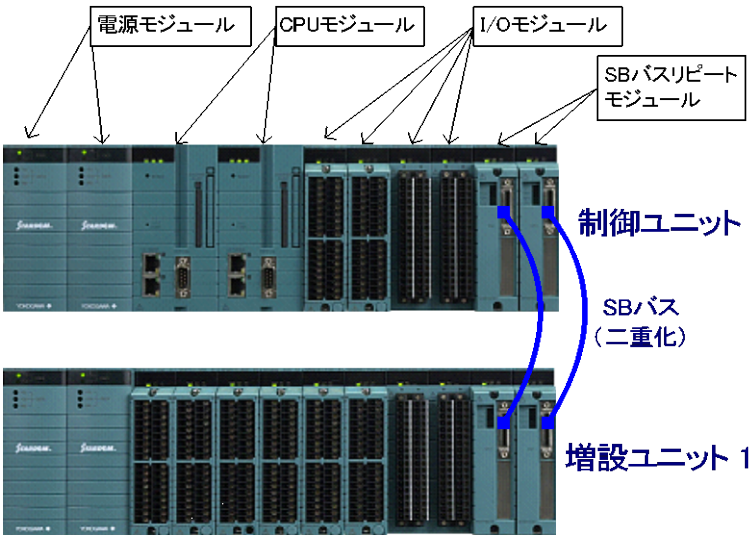


図　自律型コントローラ FCN（ベースモジュール NFBU200 使用時）

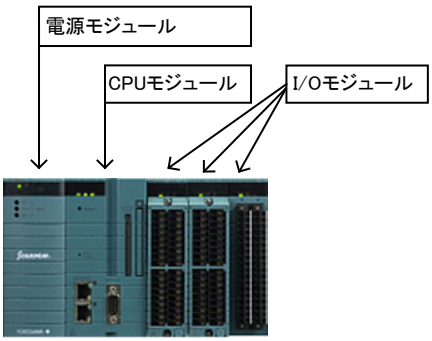


図　自律型コントローラ FCN（ショートベースモジュール NFBU050 使用時）

● モジュールの実装制限（ベースモジュール NFBU200 使用時）

ベースモジュールには、2 個の電源スロットと 10 個の I/O スロットがあります。

- CPU モジュール
  1 台の CPU モジュールは、2 個の I/O スロットを占有します。（シングル CPU の場合：1 台の CPU モジュールは、I/O スロット番号 1～2 を占有します。二重化 CPU の場合：2 台の CPU モジュールは、それぞれ、I/O スロット番号 1～2 と I/O スロット番号 3～4 を占有します。）
- SB バスリピートモジュール
  1 台の SB バスリピートモジュールは、1 個の I/O スロットを占有します。（シングル SB バスの場合：1 台の SB バスリピートモジュールは、I/O スロット番号 10 を占有します。二重化 SB バスの場合：2 台の SB バスリピートモジュールは、それぞれ、I/O スロット番号 9 と I/O スロット番号 10 を占有します。）。
- I/O モジュール
  2 台の拡張ユニットを使用した場合の I/O モジュール最大実装台数は、CPU モジュールと SB バスリピートモジュールの構成（シングル／二重化）の組み合わせ別に、下表のとおりです。

表　I/O モジュール最大実装台数

CPU モジュール	SB バスリピートモジュール	I/O モジュール実装台数
シングル	シングル	25
二重化	シングル	23
シングル	二重化	22
二重化	二重化	20

補足

I/O モジュールを実装する時には、電源モジュールの定格出力を超えないようにしてください。
I/O モジュールが消費する電源（5V DC、24V DC）については、各 GS を参照してください。

- システム電源（5V DC）の消費電流に注意が必要な I/O モジュール
  NFAV544、NFDV551、NFDV561、NFDR541
- アナログフィールド電源（24V DC）の消費電流に注意が必要な I/O モジュール
  NFAI841、NFAI143、NFAI543

● モジュールの実装制限（ショートベースモジュール NFBU050 使用時）

ショートベースモジュールには、1 台の電源モジュール、1 台の CPU モジュールおよび最大 3 台の I/O モジュールが実装されます。SB バスリピートモジュールはショートベースモジュールに実装できません。

- CPU モジュール
  1 台の CPU モジュールは、I/O スロット番号 1～2 を占有します。
- I/O モジュール
  I/O モジュールは、I/O スロット番号 3、I/O スロット番号 4 および I/O スロット番号 5 を占有します。

■ モジュールの実装例

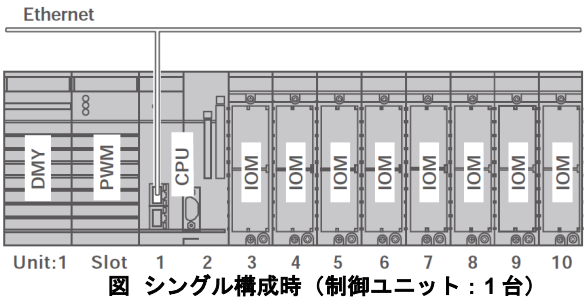


図 シングル構成時（制御ユニット：1 台）

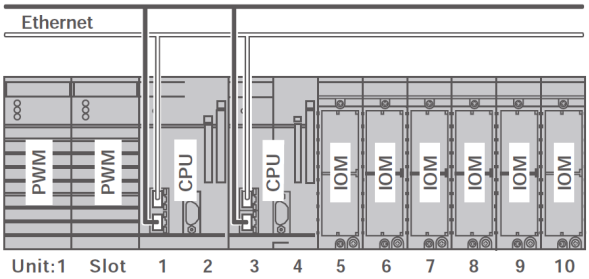


図 二重化構成時（制御ユニット：1 台）

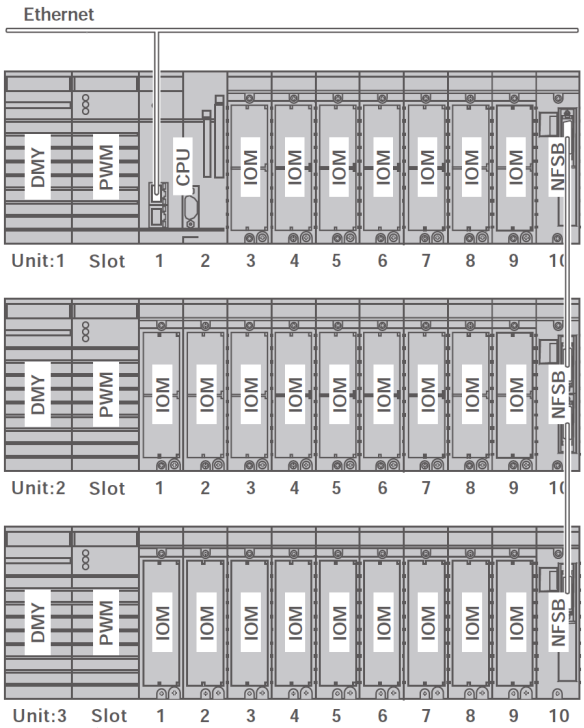


図 シングル構成時（制御ユニット:1 台＋増設ユニット:2 台）

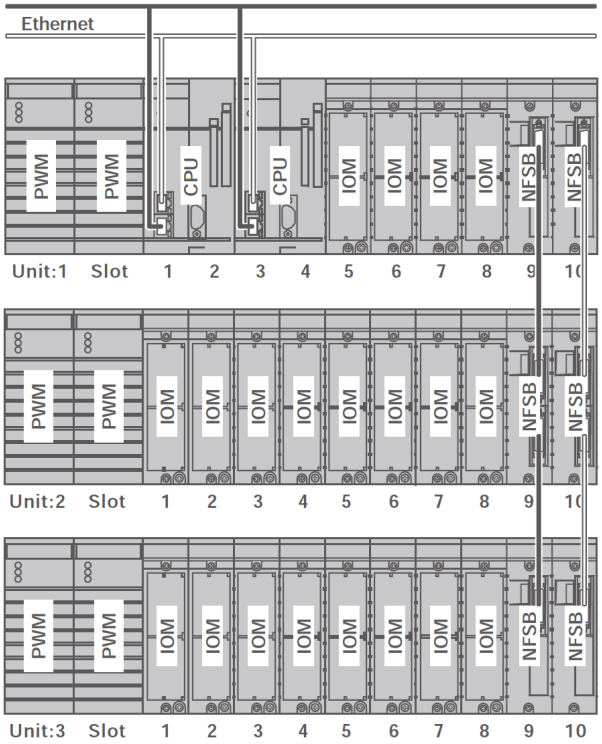


図 二重化構成時（制御ユニット:1 台＋増設ユニット:2 台）

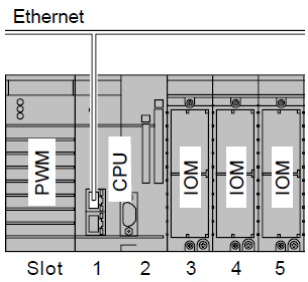


図　ショートベースモジュール

表　上図中の略号とモジュールとの対応関係

略号	モジュール
PWM	電源モジュール
CPU	CPU モジュール
IOM	I/O モジュール
NFSB	SB バスリピートモジュール
DMY	ダミーカバー（電源モジュール用）

B3.1.1 ベースモジュール NFBU200/NFBU050

本項では、ベースモジュール NFBU200 とショートベースモジュール NFBU050 の特徴を述べます。

■ ベースモジュール NFBU200

ベースモジュールには、以下の機能があります。

- **電源モジュール二重化専用スロットを装備**
- **二重化された SB バス**
- **スイッチにより、制御ユニットまたは増設ユニットに切り替え可能**
- **10 スロット実装可能（CPU モジュールは 1 モジュールで 2 スロット占有します。）**

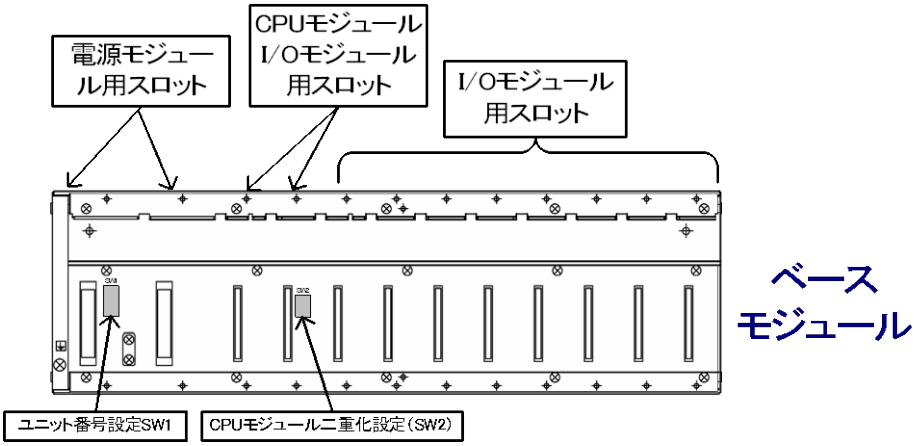


図　ベースモジュール

重 要

空きスロットには、必ずダミーカバーを装着してください。

参照

- ベースモジュールの詳細については、本書「F2. GS 集：FCN/FCJ 関連」「自律型コントローラ FCN（ハードウェア編）」（GS 34P02Q12-01）を参照してください。
- ベースモジュールの設定については、「B4.2.1 ハードウェアのセットアップ」「■ベースモジュール NFBU200 の設定」を参照してください。

表　ベースモジュール

形名	名称	仕様	質量	備考
NFBU200-S0□	FCN 用ベースモジュール（19 インチラック取付用）	・スロット構成：電源モジュール用 2 スロット、機能モジュール（CPU、I/O、SB）用 10 スロット ・占有スロット幅：CPU1 枚 2 スロット、I/O モジュール 1 枚 1 スロット、SB バスリピートモジュール 1 枚 1 スロット ・消費電流（自己消費：0.4 A（5 V）	1.9 kg	CPU シングル/二重化と制御ユニット/増設ユニットは設定にて切り替え可能
NFBU200-S1□	FCN 用ベースモジュール（DIN レール取付用）		1.0 kg	

■ ショートベースモジュール NFBU050

ショートベースモジュールには、以下の機能があります。

- **シングルの電源モジュール専用スロットを装備**
- **シングル SB バス**
- **5 I/O スロット利用可能：1 CPU モジュール（スロット番号 1～2）3 I/O モジュール（スロット番号 3、スロット番号 4、スロット番号 5）**

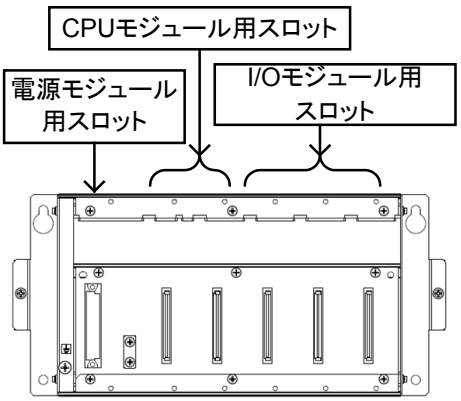


図　ショートベースモジュール

重 要

空きスロットには、必ずダミーカバーを装着してください。

参照

ショートベースモジュールの詳細については、本書「F2. GS 集：FCN/FCJ 関連」「自律型コントローラ FCN（ハードウェア編）」（GS 34P02Q12-01）を参照してください。

表　ショートベースモジュール

形名	名称	仕様	質量	備考
NFBU050-S1□	ショートベースモジュール（DIN レール取付用）	・スロット構成：電源モジュール用 1 スロット、機能モジュール（CPU、I/O）用 5 スロット ・占有スロット幅：CPU1 枚 2 スロット、I/O モジュール 1 枚 1 スロット、 ・消費電流（自己消費：0.025 A（5 V）	0.58 kg	

B3.1.2 電源モジュール（FCN 用）

電源モジュールは、各ベースモジュールに実装され、安定した電源を供給します。さらに各ベースモジュールで二重化が可能です。
また、アナログフィールド電源（24V DC）用の入力端子を備えており、この端子よりベースモジュールを経由し各 I/O モジュールへ供給されます。デジタル出力モジュールにもフィールド電源（24V DC）が必要な I/O モジュールがあります。この場合は、I/O モジュール端子に個別に供給する必要があります。
電源モジュールは、活線挿抜（ホットスワップ）も可能です。

参照

- I/O モジュール端子の詳細については、本書「F2. GS 集：FCN/FCJ 関連」で各 I/O モジュールの仕様を参照してください。
- 電源モジュールの詳細については、本書「F2. GS 集：FCN/FCJ 関連」「自律型コントローラ FCN（ハードウェア編）」（GS 34P02Q12-01）を参照してください。

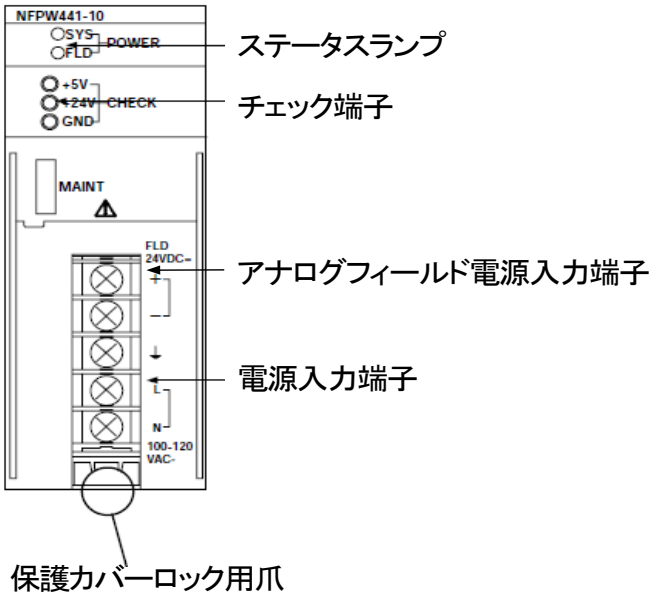


図　電源モジュールの外観

■ 電源モジュールの二重化

二重化された電源は、CPU モジュールおよび各 I/O モジュールに供給されます。
この供給された電源は、突き合わせダイオードにより内部回路に取り込まれます。

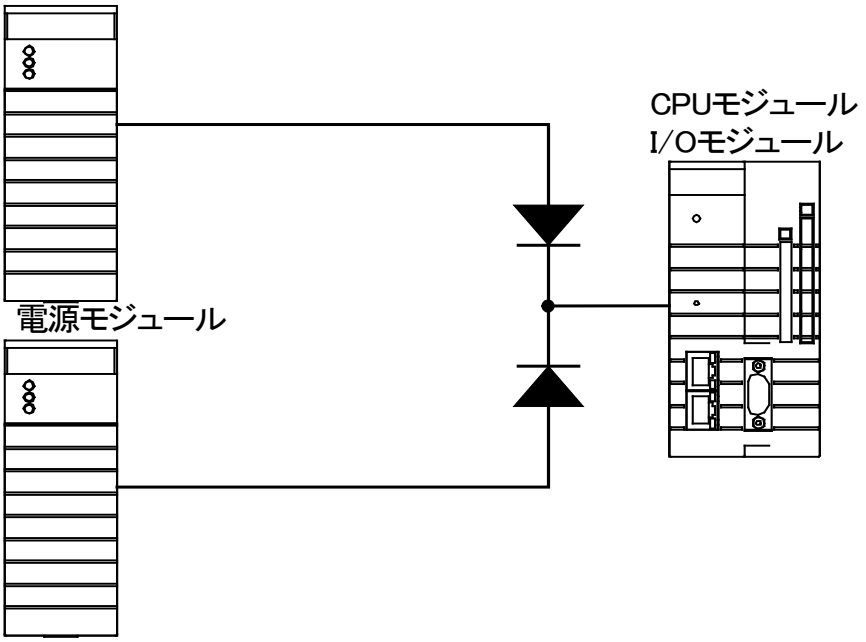


図　二重化された電源モジュールの電源供給

■ 電源モジュール（FCN 用）

表　電源モジュール（FCN 用）

形名	名称	仕様	備考
NFPW441	FCN 用電源モジュール（100～120VAC 入力）	・二重化：可能、ホットスワップ（活線挿抜）：可能 ・定格出力：+5.1VDC、7.8A ・アナログフィールド電源中継機能：入力：24VDC±10%、4A、二重化対応（ダイオード突合せ） ・表示：SYS-POWER（+5VDC 供給中）、FLD-POWER（アナログフィールド電源供給中） ・出力電圧チェック端子（+5V、+24V） ・質量：0.6kg	・電源シングル/二重化共用（ベースモジュールに 2 枚装着して二重化として使用可能）
NFPW442	FCN 用電源モジュール（220～240VAC 入力）		
NFPW444	FCN 用電源モジュール（24VDC 入力）		

参照

電源モジュールの停電不感時間については、「B7.4.1 停電不感時間」を参照してください。

B3.1.3 CPU モジュール（FCN 用）

CPU モジュールは、制御ユニットに 1 台（二重化の場合には 2 台）実装されます。
OS としてリアルタイム OS を採用し、国際標準である IEC 61131-3 に準拠した制御言語をサポートしています。さらに Java アプリケーションも動作します。
CPU モジュール（ハードウェア）には、以下の特長があります。

● CPU モジュール（ハードウェア）の特長

- CPU を二重化でき、活線挿抜（ホットスワップ）が可能です。
- メモリに ECC（Error Correcting Code）を採用しています。
- ファンレス設計です。
- RAS 機能（CPU 自己診断、温度監視、I/O 診断など）が充実しています。

■ 外観図

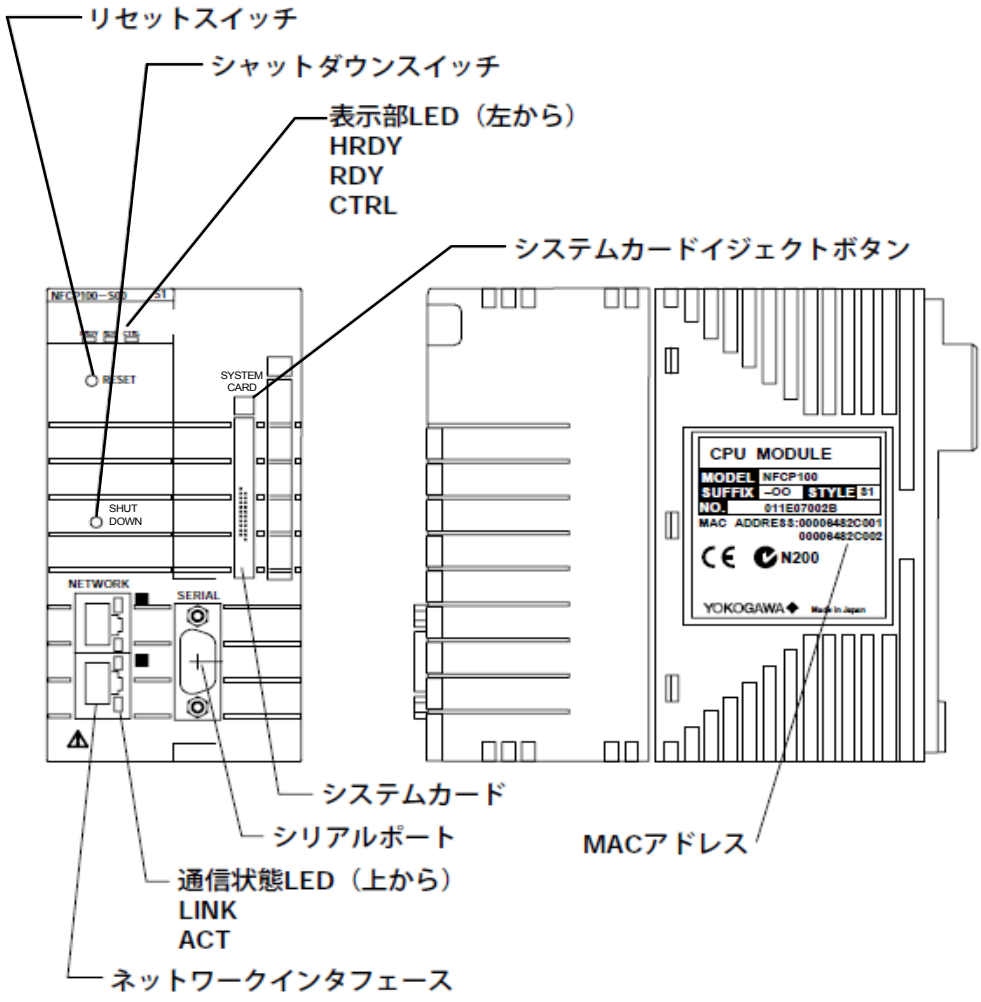


図　CPU モジュール（FCN 用）の外観

■ CPU モジュール（FCN 用）

表　CPU モジュール（FCN 用）

項目		仕様
形名		NFCP100-S0□
CPU		MMX-Pentium 166 MHz
メモリ	メイン	128 MB、ECC 機能付き
	保持用	1 MB、ECC 機能付き、バッテリバックアップ
システムカード		1 スロット
シリアルポート(*2)		RS-232-C：1 ポート D-Sub9 ピン（オス)(*1)
	通信方式	全二重方式
	同期方式	調歩同期式
	通信速度	0.3、1.2、2.4、4.8、9.6、14.4、19.2、28.8、38.4、57.6、115.2 kbps
ネットワーク インタフェース		100/10 Mbps（100BASE-TX、10BASE-T） Ethernet 2 ポート、RJ-45 モジュラージャック
I/O インタフェース		SB バス（二重化）
RAS 機能		WDT、温度監視など
バッテリ		Li 電池 2700 mAh
表示		CPU 状態表示（3LED） LAN 状態表示（2LED×2 ポート）
スイッチ		リセット、シャットダウン
電源	供給電圧	5 V DC±5 %
	消費電流	1800 mA(max)
二重化		可能
質量		0.7 kg
外形	寸法（W/H/D）	65.8×132×145.7 mm
	占有スロット	2 スロット占有

*1:　コネクタを止めるネジは、インチネジ（No.4-40UNC）です。
*2:　CPU モジュール二重化構成時は、使用できません。

参照

CPU モジュールの詳細については、本書「F2. GS 集：FCN/FCJ 関連」「自律型コントローラ FCN（ハードウェア編）」（GS 34P02Q12-01）を参照してください。

■ CPU の二重化

CPU モジュール、ベースモジュールは、シングル／二重化兼用です。このため、制御対象に必要な信頼性レベルに応じて、CPU モジュール 1 枚の追加により二重化システムを構成できます。二重化のための専用品は、必要有りませんので補用品の負担を少なくできます。
CPU モジュール故障時には、故障した CPU モジュールからシステムカードを新しい CPU モジュールに移し替え、CPU モジュールを交換するだけで済み、パソコンツールなどは不要です。
また、制御アプリケーションを作成する場合も、CPU モジュールの二重化を意識する必要はありません。

重 要

CPU モジュールを二重化した場合、CPU モジュールのシリアルポートによる通信、および Java アプリケーションは使用できません。

補足

ソフトウェアライセンスとして「FCN 基本ソフトウェアライセンス 二重化用」が、別途必要です。

参照

- 二重化された CPU モジュールの詳細動作については、「B7.2 CPU の二重化」を参照してください。
- 二重化した CPU モジュールの制御アプリケーションを作成およびデバッグなどにおける注意事項については、「B4.2.4 制御アプリケーション作成上の注意」を参照してください。

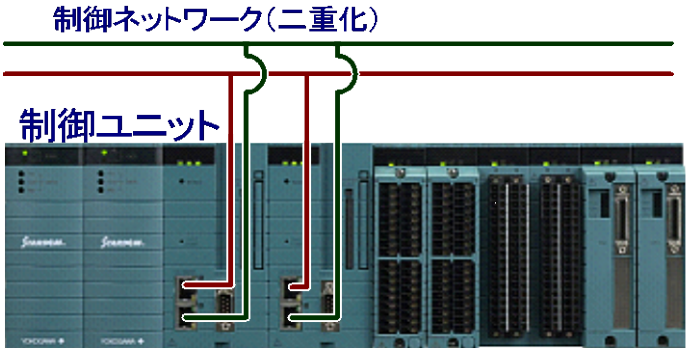


図 CPU モジュールを二重化した例

■ FCN/FCJ システムカード

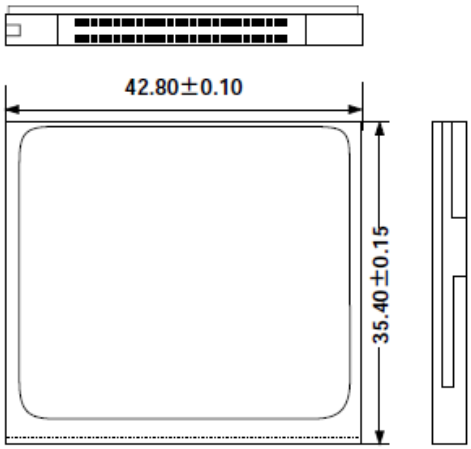
FCN/FCJ システムカードは、FCN/FCJ が動作するために必要な情報がすべて保存されています。
システムカードに保存されている主な情報には、以下のものがあります。

- OS／各種ドライバ
- 制御実行エンジン
- 制御アプリケーション
- ネットワーク設定情報
- I/O 設定情報
- Java 実行環境
- Java アプリケーション

FCN/FCJ システムカードは、FCN/FCJ の専用スロットに実装して使用します。

参照

FCN/FCJ システムカードは、FCN/FCJ 基本ソフトウェアライセンスを購入するとオーダ ID シートと合わせて提供されます（FCN 二重化用の基本ソフトライセンスは、FCN/FCJ システムカードが 2 枚提供されます）。FCN/FCJ 基本ソフトウェアライセンスの詳細については、「B5.3 FCN/FCJ 動作に必要なソフトウェア」を参照してください。



FCN/FCJ	
SYSTEM CARD	
MODEL	NT225AA
SUFFIX	-0□
REV.	
NO.	

YOKOGAWA ◆ Made in japan

［ラベル］

図　FCN/FCJ システムカードの外観

重 要

- FCN/FCJ の通電中は、FCN/FCJ システムカードを着脱しないでください。
- 端子部に触れないでください。
- 熱、水分、直射日光を避けてください。
- 衝撃を与えないでください。

B3.1.4 SB バスリピートモジュール（FCN 用）

SB バスリピートモジュールは、制御ユニットと I/O モジュール増設ユニットとを接続するためのモジュールです。
SB バスを二重化する場合には、各ユニットに SB バスリピートモジュールを 2 枚ずつ実装します。各 SB バスリピートモジュールは、専用分岐栓を介して、専用ケーブルで接続します。

重 要

NFSB100（FCN 用 SB バスリピートモジュール）は、NFBU050（ショートベースモジュール）に実装できません。

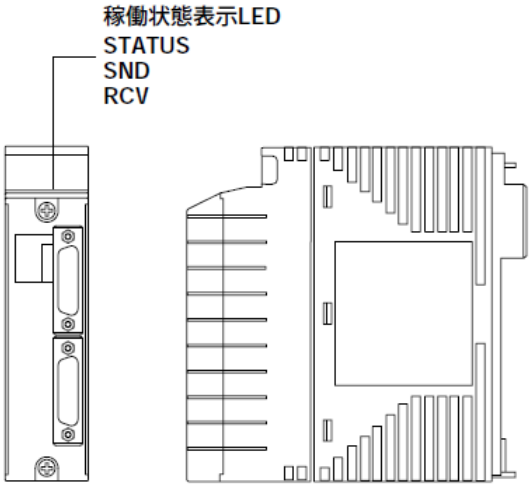


図 SB バスリピートモジュール外観図

■ SB バスリピートモジュール（FCN 用）

表 SB バスリピートモジュール（FCN 用）

形名	名称	仕様	備考
NFSB100	FCN 用 SB バスリピートモジュール	・二重化：可能、ホットスワップ（活線挿抜）：可能 ・伝送方式：シリアル通信（128Mbps）、 ・伝送距離：1 系統の総ケーブル長 8m ・増設ユニット：最大 2 ユニット（合計 3 ユニット） ・表示：STATUS（ハードウェア正常状態）、SND（送信中）、RCV（受信中） ・消費電流：500mA（5VDC）、質量：0.2kg、	SB シングル/二重化共用（ベースモジュールに 2 枚装着して二重化として使用可能）

■ SB バスの二重化

FCN のベースモジュール上の SB バスは、二重化されています。さらに増設ユニットを接続するときに、SB バスリピートモジュールの二重化により、増設ユニットの SB バスも二重化できます。
SB バスが二重化されていると、それぞれの系を 1 秒周期で交互に使用します。何らかの原因により SB バスがシングル状態になった場合には、正常に動作している SB バスを優先して使用します。SB バスの二重化が復帰ししだい、二重化動作が自動的に再開されます。

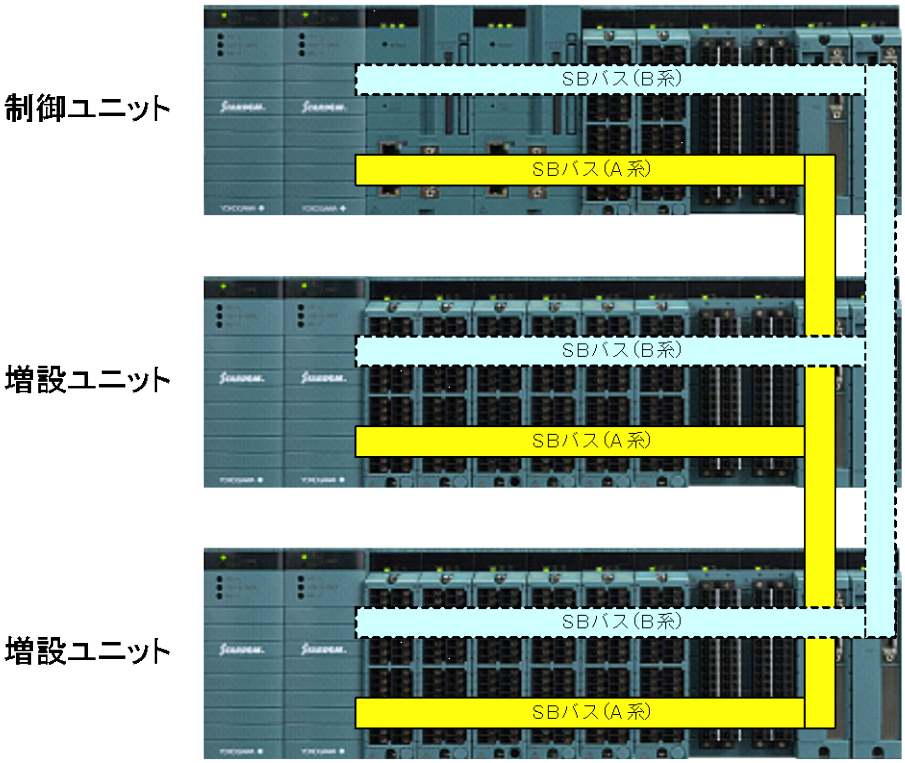


図　二重化された SB バス

B3.1.5 I/O モジュール

自律型コントローラ FCN では、豊富な I/O モジュールをサポートしています。

参照

詳細については、下記の GS を参照してください。
- アナログ入出力モジュール（GS 34P02Q31-01）（本書「F2. GS 集：FCN/FCJ 関連」）
- デジタル入出力モジュール（GS 34P02Q35-01）（本書「F2. GS 集：FCN/FCJ 関連」）
- NFLF111 Foundation Communication Module（GS 34P02Q55-01E）（英文版のみにつき本書には付属していません）
- シリアル通信モジュール（GS 34P02Q36-01）（本書「F2. GS 集：FCN/FCJ 関連」）
- 端子ブロック（GS 34P02Q41-01）（本書「F2. GS 集：FCN/FCJ 関連」）
- MIL コネクタ端子台／ケーブル／リレーターミナルケーブル（GS 34P02Q43-01）（本書「F2. GS 集：FCN/FCJ 関連」）

■ アナログ入出力モジュール（FCN 用）

B

表　アナログ入出力モジュール（FCN 用）（非絶縁モジュール）

形名	名称	絶縁方式	押締 (*1)	MIL (*2)	仕様	備考
NFAI141	アナログ入力 モジュール （4～20mA 入力、16 点）	非絶縁	○	40	・精度:フルスケールの±0.1% ・データ更新周期:10ms、入力ステップ応答時間:100ms ・温度ドリフト:±0.01%/℃（Max）	(*3)
NFAV141	アナログ入力モジュール （1～5V 入力、16 点）	非絶縁	○	40	・精度:フルスケールの±0.1% ・データ更新周期:10ms、入力ステップ応答時間:100ms ・温度ドリフト:±0.01%/℃（Max）	－
NFAV142	アナログ入力モジュール （－10～＋10V 入力、 16 点）	非絶縁	○	40	・精度:フルスケールの±0.1% ・データ更新周期:10ms、入力ステップ応答時間:100ms ・温度ドリフト:±0.01%/℃（Max）	－
NFAI841	アナログ入出力モジュール （4～20mA 入力、8 点 4～20mA 出力、8 点）	非絶縁	○	40	・出力フォールバック設定：チャネルごと ・入力精度:フルスケールの±0.1% ・出力精度:フルスケールの±0.3% ・データ更新周期:10ms、　入力ステップ応答時間:100ms、 出力ステップ応答時間:40ms ・温度ドリフト:±0.01%/℃（Max）	(*3)
NFAB841	アナログ入出力モジュール （1～5V 入力、8 点 4～20mA 出力、8 点）	非絶縁	○	40	・入力：差動（許容コモンモード±1V 以下） ・出力フォールバック設定：チャネルごと ・入力精度:フルスケールの±0.1% ・出力精度:フルスケールの±0.3% ・データ更新周期:10ms、入力ステップ応答時間:100ms、 出力ステップ応答時間:40ms ・温度ドリフト:±0.01%/℃（Max）	－
NFAV542	アナログ出力モジュール （－10V～＋10V 出力、 16 点）	非絶縁	○	40	・出力フォールバック設定：チャネルごと ・出力精度:フルスケールの±0.3% ・データ更新周期:10ms、出力ステップ応答時間:40ms ・温度ドリフト:±0.01%/℃（Max）	－

*1:　○：押し締め端子ブロック有り（サージアブソーバ内蔵タイプも有り）、×：押し締め端子台接続不可
*2:　MIL コネクタ接続、40：40 ピン MIL コネクタ接続可、×：MIL コネクタ接続不可
*3:　2 線式/4 線式伝送器設定：チャネルごと（設定ピン）

共通仕様
・表示：STATUS（ハードウェア正常）、ACT（入出力動作中）
・ホットスワップ（活線挿抜）：可能

表　アナログ入出力モジュール（FCN 用）（一括絶縁モジュール）

形名	名称	絶縁方式	押締 (*1)	MIL (*2)	仕様	備考
NFAI143	アナログ入力モジュール（4～20mA 入力、16 点）	一括絶縁	○	40	・耐電圧：1500VAC（入力/システム間） ・入力精度:フルスケールの±0.1% ・データ更新周期:10ms、入力ステップ応答時間:100ms ・温度ドリフト:±0.01%/℃（Max）	(*3)
NFAV144	アナログ入力モジュール（1～5V 入力/－10V～＋10V 入力、16 点）	一括絶縁	○	40	・入力切替：1～5V/－10～＋10V ・耐電圧：1500VAC（入力/システム間） ・入力精度:フルスケールの±0.1% ・データ更新周期:10ms、入力ステップ応答時間:100ms ・温度ドリフト:±0.01%/℃（Max）	(*4)
NFAI543	アナログ出力モジュール（4～20mA 出力、16 点）	一括絶縁	○	40	・耐電圧：1500VAC（出力/システム間） ・出力フォールバック設定：チャネルごと ・出力精度:フルスケールの±0.3% ・データ更新周期:10ms、出力ステップ応答時間:100ms ・温度ドリフト:±0.01%/℃（Max）	－
NFAV544	アナログ出力モジュール（－10V～＋10V 出力、16 点）	一括絶縁	○	40	・耐電圧：1500VAC（出力/システム間） ・出力フォールバック設定：チャネルごと ・出力精度:フルスケールの±0.3% ・データ更新周期:10ms、出力ステップ応答時間:40ms ・温度ドリフト:±0.01%/℃（Max）	－
NFAT141	熱電対/mV 入力モジュール（熱電対/mV 入力、16 点）	一括絶縁	○	40 *5	・入力信号切り替え：各点ごと設定 ・バーンアウト検出：可能、検出時間 60s ・耐電圧：1500VAC（入力/システム間） ・TC 入力精度:フルスケール（－20～80mV）の±0.03% ・mV 入力精度:フルスケール（－100～150mV）の±0.032% ・データ更新周期:1s・TC 入力温度ドリフト:±30ppm/℃（Max） ・mV 入力温度ドリフト:±32ppm/℃（Max）	－
NFAR181	測温抵抗体入力モジュール（測温抵抗体入力、12 点）	一括絶縁	○	×	・入力信号切り替え：各点ごと設定 ・バーンアウト検出：可能、検出時間 60s ・耐電圧：1500VAC（入力/システム間） ・入力精度:フルスケール（0～400Ω）の±0.03% ・データ更新周期:1s ・温度ドリフト:±30ppm/℃（Max）	－

*1:　○：押し締め端子台有り（サージアブソーバ内蔵タイプも有り）、×：押し締め端子台接続不可
*2:　MIL コネクタ接続、40：40 ピン MIL コネクタ接続可、×：MIL コネクタ接続不可
*3:　2 線式/4 線式伝送器設定：チャネルごと（設定ピン）
*4:　入力信号切り替え（ツール）－10～+10V 選択時チャネルごとレンジ設定可
*5:　MIL コネクタ接続の場合は、mV 入力としてのみ使用可能

共通仕様
・表示：STATUS（ハードウェア正常）、ACT（入出力動作中）
・ホットスワップ（活線挿抜）：可能

表　アナログ入出力モジュール（FCN 用）（個別絶縁モジュール）

形名	名称	絶縁方式	押締 (*1)	MIL (*2)	仕様	備考
NFAI135	アナログ入力モジュール（4～20mA、8 点）	個別絶縁	○	40	・耐電圧：500VAC（入力/システム間）、500VAC（チャネル間） ・入力精度:フルスケールの±0.1% ・データ更新周期:10ms ・入力ステップ応答時間:100ms ・温度ドリフト:±0.01%/℃（Max）	(*3)
NFAI835	アナログ入出力モジュール（4～20mA 入力、4 点 4～20mA 出力、4 点）	個別絶縁	○	40	・耐電圧：500VAC（入出力/システム間）、500VAC（チャネル間） ・出力フォールバック設定：チャネルごと ・入力精度:フルスケールの±0.1% ・出力精度:フルスケールの±0.3% ・データ更新周期:10ms、入力ステップ応答時間:100ms、出力ステップ応答時間:100ms ・温度ドリフト:±0.01%/℃（Max）	(*3)
NFAP135	パルス入力モジュール（パルス(0～10kHz)入力、8 点）	個別絶縁	○	40	・耐電圧：500VAC（入力/システム間）、500VAC（チャネル間） ・最小入力パルス幅:40μs ・データ更新周期:2ms ・入力可能パルス形式：下記 5 種 無電圧接点パルス（オープンコレクタ/リレー） 電圧パルス電源供給電流パルス（2 線式/3 線式）	(*4)
NFAF135	周波数入力モジュール（パルス（0.1Hz～10kHz）入力、8 点）	個別絶縁	○	40	・耐電圧：500VAC（入力/システム間）、500VAC（チャネル間） ・最小入力パルス幅:40μs ・データ更新周期:10ms ・入力可能パルス形式：下記 3 種 無電圧接点パルス（オープンコレクタ/リレー） 電圧パルス	(*4)

*1:　○：押し締め端子台有り（サージアブソーバ内蔵タイプも有り）、×：押し締め端子台接続不可
*2:　MIL コネクタ接続、40：40 ピン MIL コネクタ接続可、×：MIL コネクタ接続不可
*3:　2 線式/4 線式伝送器設定：チャネルごと（接続端子選択）
*4:　入力信号切り替え（ツールおよび接続端子）選択にてチャネルごと

共通仕様
・表示：STATUS（ハードウェア正常）、ACT（入出力動作中）
・ホットスワップ（活線挿抜）：可能

■ デジタル入出力モジュール（FCN 用）

表　デジタル入出力モジュール（FCN 用）

形名	名称	押締 (*1)	MIL (*2)	仕様	備考
NFDV151	デジタル入力モジュール （24VDC 接点入力、32 点）	○	50	・機能：ステータス/押しボタン（エッジカウント） ・入力応答時間：8ms 以下（ステータス入力時） ・最小 ON 検出時間：20ms（押しボタン入力時） ・最大 ON/OFF 周期：25Hz（押しボタン入力時）	(*3)
NFDV157	デジタル入力モジュール （24VDC 接点入力、32 点）	○	×	・機能：ステータス ・入力応答時間：8ms 以下（ステータス入力時） ・最小 ON 検出時間：20ms（押しボタン入力時） ・最大 ON/OFF 周期：25Hz（押しボタン入力時）	(*3)
NFDV161	デジタル入力モジュール （24VDC 接点入力、64 点）	×	50 2 組	・機能：ステータス/押しボタン（エッジカウント）(*8) ・入力応答時間：8ms 以下（ステータス入力時） ・最小 ON 検出時間：20ms（押しボタン入力時） ・最大 ON/OFF 周期：25Hz（押しボタン入力時）	(*3)
NFDV141	デジタル入力モジュール （100～120VAC 接点入力、16 点）	○	×	・機能：ステータス/押しボタン（エッジカウント） ・入力応答時間：160ms 以下（ステータス入力時） ・最小 ON 検出時間：200ms（押しボタン入力時） ・最大 ON/OFF 周期：2.5Hz（押しボタン入力時）	(*4)
NFDV142	デジタル入力モジュール （200～240VAC 接点入力、16 点）	○	×	・機能：ステータス/押しボタン（エッジカウント） ・入力応答時間：160ms 以下（ステータス入力時） ・最小 ON 検出時間：200ms（押しボタン入力時） ・最大 ON/OFF 周期：2.5Hz（押しボタン入力時）	(*4)
NFDV532	パルス幅出力モジュール （4 チャンネル：Up パルス/ Down パルス各 1 点）	○	50	・出力フォールバック設定：全チャネル一括設定 ・パルス出力精度：最小 2ms、2ms 単位 （誤差：±1ms Max)	(*5)
NFDV551	デジタル出力モジュール （24VDC 接点出力、32 点）	○	50	・出力フォールバック設定：全チャネル一括設定 ・出力応答時間：3ms 以下（ステータス出力時）	(*6)
NFDV557	デジタル出力モジュール （24VDC 接点出力、32 点）	○	×	・出力フォールバック設定：全チャネル一括設定 ・出力応答時間：3ms 以下（ステータス出力時）	(*6)
NFDV561	デジタル出力モジュール （24VDC 接点出力、64 点）	×	50 2 組	・出力フォールバック設定：全チャネル一括設定 ・出力応答時間：3ms 以下（ステータス出力時）	(*6)
NFDR541	リレー出力モジュール （24～110VDC/100～240VAC リレー出力、16 点）	○	×	・印加電圧：24～110VDC、100～240VAC ・出力フォールバック設定：全チャネル一括設定 ・出力応答時間：12ms 以下（ステータス出力時）	(*7)

*1:　○：押し締め端子台有り、×：押し締め端子台接続不可
*2:　MIL コネクタ接続、 50：50 ピン MIL コネクタ接続可、×：MIL コネクタ接続不可
*3:　16 点ごとにコモン（プラスコモン/マイナスコモン対応）
*4:　8 点ごとにコモン
*5:　コモン共通（マイナスコモン）
*6:　16 点ごとにコモン（マイナスコモン）
*7:　8 点ごとにコモン（プラスコモン/マイナスコモン対応）
*8:　押しボタン入力機能は入力 1～32 まで可能

共通仕様
・表示：STATUS（ハードウェア正常）、ACT（入出力動作中）
・ホットスワップ（活線挿抜）：可能

■ 通信モジュール（FCN 用）

表　通信モジュール（FCN 用）

形名	名称	仕様	備考
NFLF111	Foundation fieldbus 通信モジュール (*1)	・ポート数：4 ・通信速度：31.25kbps	最大 8 組実装可 (*2) 接続：専用押し締め端子ブロック（形名:NFTF9S）
NFLR111	シリアル通信モジュール（RS-232-C）	・ポート数：2 ・通信速度：0.3～115.2kbps	接続：D-Sub9 ピン（メス×2） 固定：ミリネジ（M2.6）
NFLR121	シリアル通信モジュール（RS-422/RS-485）	・ポート数：2 ・通信速度：0.3～115.2kbps	接続：端子台 M4 ネジ止め（5 極×2）
NFLC121	CANopen 通信モジュール	・ポート数：1 ・通信速度：10kbps～1Mbps	接続：D-Sub9 ピン（オス） 固定：インチネジ（No.4-40UNC）
NFLP121	PROFIBUS-DP 通信モジュール	・ポート数：1 ・通信速度：9.6kbps～12Mbps	接続：D-Sub9 ピン（メス） 固定：インチネジ（No.4-40UNC）

*1:　FOUNDATION fieldbus 機能を定義するツールは、STARDOM においては英文版のみです。
*2:　NFLF111 を 8 組実装した場合、他の I/O モジュールは 4 枚まで実装できます。
NFLF111 を 7 組実装した場合、他の I/O モジュールは 16 枚まで実装できます。
NFLF111 を 6 組以下の場合、I/O モジュールの実装制限はありません。

B3.2 自律型コントローラ FCJ

FCJ は、I/O インタフェースを内蔵したオールインワンタイプの高機能、高信頼コントローラで、FCN と同一のアプリケーションを実行できます。
FCJ（ハードウェア）には、以下の特長があります。

●FCJ（ハードウェア）の特長

- メモリに ECC（Error Correcting Code）を採用しています。
- ファンレス設計です。
- RAS 機能（CPU 自己診断、温度監視、I/O 診断など）が充実しています。
- 制御ネットワークを二重化できます。

■ FCJ の外観

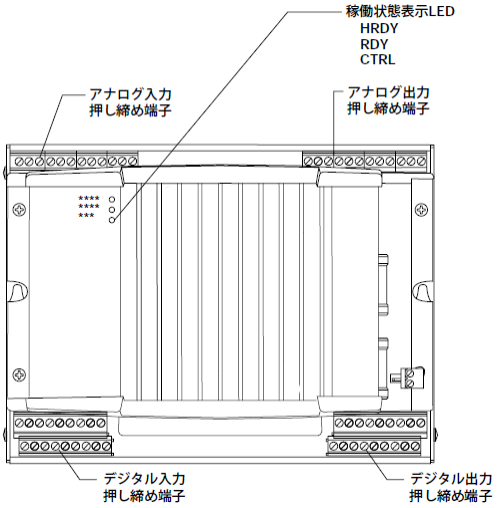


図　FCJ の外観（正面）

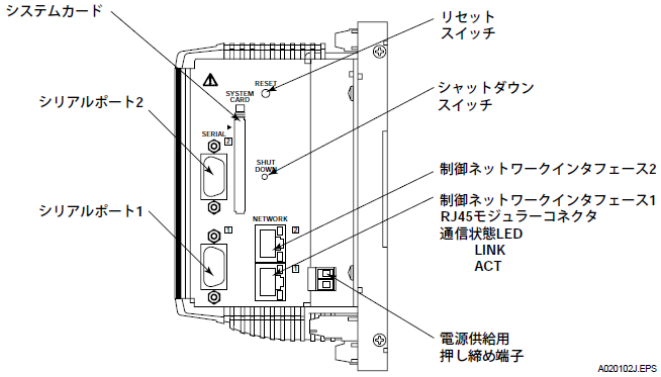


図　FCJ の外観（左側面）

■ 自律型コントローラ FCJ

表　FCJ

形名	名称	基本仕様	仕様	備考
NFJT100	自律型コントローラ FCJ	・表示（稼働状態）： HRDY（ハードウェア正常状態）、 RDY（システム正常状態）、 CTRL（制御動作可能状態） ・表示（ネットワークインタフェース）： LINK、ACT ・電源：24VDC±10%、 消費電流：800mA（24VDC）(*1) ・質量：1.7kg ・外形寸法（W/H/D）：213.8×156×95mm	・Ethernet：2 ポート ・RS-232-C：2 ポート ・AI（1～5V）：6 点 ・AO（4～20mA）：2 点 ・DI（24VDC）：16 点 ・DO（24VDC）：16 点 ・FF-H1：2 セグメント (*2)	・CPU 部仕様： NFCP100 と同等 ・デジタル入出力： 各 16 点一括コモン ・設置：盤取り付けまたは DIN レール取り付け

*1:　NFJT100-H10□（FOUNDATION fieldbus 機能付き）の消費電流は、1、000mA（24V DC）です。
*2:　オプションにより実装可能。FOUNDATION fieldbus 機能を定義するツールは、STARDOM においては英文版のみです。

B4. FCN/FCJ のソフトウェア

■ FCN/FCJ 内部構成

FCN/FCJ は、下記のようなソフトウェア構成で動作します。
ユーザが作成するアプリケーションは、この中の
・ 制御アプリケーション
・ Java アプリケーション
です。
本章では、これらを開発するツールおよび開発手順、パフォーマンスの見積もりなどを説明します。

表　FCN/FCJ の内部構成

ソフトウェア	機能	備考
リアルタイム OS プロセス入出力	システムカード、制御ネットワーク、I/O などの管理・制御を実施	OS 部
制御実行エンジン	制御アプリケーションを実行するプラットフォーム	プラットフォーム部
Java プラットフォーム	Java アプリケーションを実行するプラットフォーム	プラットフォーム部
制御アプリケーション	ユーザが作成するシーケンス制御やループ制御プログラム	ユーザアプリケーション部
Java アプリケーション	InfoWell やユーザが作成する Java アプリケーションプログラム	ユーザアプリケーション部

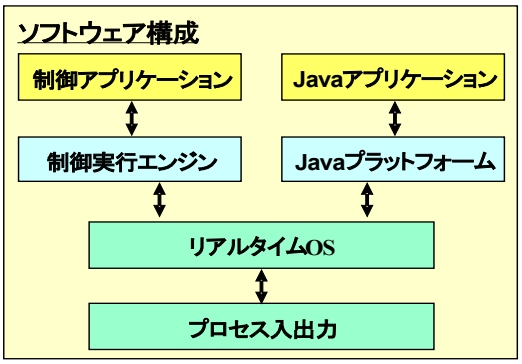


図　FCN/FCJ の内部構成

B4.1 開発環境／ツール

「制御アプリケーション」や「Java アプリケーション」を開発するには、それぞれ、専用の開発ツールが必要です。

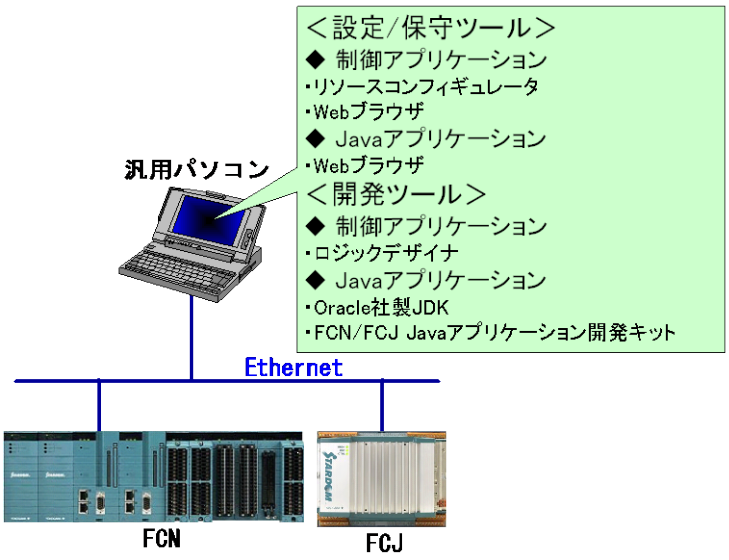


図　FCN/FCJ の開発環境

● 制御アプリケーション

FCN/FCJ の開発および保守には、Ethernet で接続した汎用パソコンを使います。
FCN/FCJ の IP アドレス、各 I/O モジュール設定、動作させるソフトウェアのライセンス登録などの基本設定には、汎用パソコン上のリソースコンフィギュレータを用います。
FCN/FCJ の情報発信機能の基本設定や、日付／時刻設定、ログファイルの表示などの操作には、汎用パソコン上の Web ブラウザを用います。
FCN/FCJ 上で動作する制御アプリケーションの作成・構築／デバッグ／ダウンロードには、ロジックデザイナを用います。

参照

情報発信機能については、「B2.3 FCN/FCJ の自律発信機能」を参照してください。

表　制御アプリケーション開発環境

項目	仕様	
パソコン	PC/AT 互換機	
CPU	Windows 7 (32bit)	1GHz 以上の 32bit(x86)プロセッサまたは 64bit(x64)プロセッサ
	Windows 7 (64bit)	1GHz 以上の 64bit(x64)プロセッサ
メモリ	2 GB 以上	
ハードディスク	20 GB 以上の空き領域	
ディスプレイ	1024×768 以上、True Color	
Ethernet アダプタ	下記 OS で動作するもの	
DVD-ROM ドライブ	インストール時のみ必要（下記 OS で動作するもの）	
OS	Windows 7 Professional SP1 (32bit/64bit)	

● Java アプリケーション

Java アプリケーションは、FCN/FCJ Java アプリケーション開発キットと Oracle 社製 JDK（Java Development Kit、JDK 1.4.2）を組み合わせて開発します。
FCN/FCJ Java アプリケーション開発キットでは、FCN/FCJ 上の Java アプリケーションを開発する上で必要となるクラスライブラリ（制御機能データアクセス用／Java システム管理用）や、デバッグを支援するツールを提供します。
また、Web ブラウザが、FCN/FCJ Java 機能の動作環境を設定するために必要です。
さらに Web 監視アプリケーションを容易に作成できるオプションパッケージ（Webmetry 基本ライブラリポートフォリオ、InfoWell 情報発信パッケージ）も用意しています。

参照

Java アプリケーション開発環境の詳細については、「B4.1.5 FCN/FCJ Java アプリケーション開発」を参照してください。

B

■ 開発ツール／ソフトウェア部品とその機能

開発ツール／ソフトウェア部品とその機能を下表に示します。

表　開発ツール／ソフトウェア部品とその機能（一覧）

開発ツール	機能	備考
リソースコンフィギュレータ	FCN/FCJ の基本設定ツール。主に以下の機能があります。 ・実 I/O と論理 I/O の連結設定 ・ネットワーク設定 ・ライセンス設定 ・CPU モジュール設定 ・I/O モジュール設定 ・RAS 情報の表示	B4.1.1 章参照
Web ブラウザ	FCN/FCJ への詳細情報設定ツール・FCN/FCJ の操作ツール	B4.1.6 章参照
ロジックデザイナ	制御アプリケーションを作成／デバッグするツール。 以下の特長がある ・国際標準規格 IEC 61131-3 に準拠した 5 種類の言語をサポート ・制御ループ構築機能によりプログラミングレスで、簡単に制御アプリケーション構築可能 ・強力なデバッグ機能を装備	B4.1.2 章参照
ファンクション／ファンクションブロック	ロジックデザイナに標準で装備されている制御アプリケーションを作成するための部品	B4.1.3 章参照
アプリケーションポートフォリオ（APPF）	アプリケーションノウハウをライブラリ化した制御アプリケーションを作成するためのオプション部品	B4.1.4 章参照
FCN/FCJ Java アプリケーション開発キット	FCN/FCJ の Java アプリケーションを開発必要な部品	B4.1.5 章参照
Oracle 社製 JDK（Java Development Kit）	Java アプリケーション開発環境	B4.1.5 章参照

B4.1.1 リソースコンフィギュレータ

リソースコンフィギュレータには、FCN/FCJ の基本設定など、以下の機能があります。

● 物理 I/O と論理 I/O の連結設定

ロジックデザイナで作成する制御アプリケーションは、すべて論理 I/O（デバイスラベル変数）で定義されます。リソースコンフィギュレータでは、この論理 I/O（デバイスラベル変数）を物理 I/O（デバイスラベル）に割り当てます。

補足

- 制御アプリケーションは、物理 I/O に左右されずに開発できます。これにより、移植しやすい制御アプリケーションを開発でき、再利用による生産性を高められます。
- 物理 I/O（デバイスラベル）にデフォルトで定義されている論理 I/O（デバイスラベル変数）を用いても、制御アプリケーションを作成できます。

参照

論理 I/O（デバイスラベル変数）や物理 I/O（デバイスラベル）の結合に関する詳細は、「B4.3.4 入出力処理」「■デバイスラベルとデバイスラベル変数」を参照してください。

● ネットワーク設定

コントローラのネットワークに関する属性（IP アドレス、サブネットマスク、デフォルトゲートウエイ）を設定する機能です。
コントローラの工場出荷時には IP アドレスは設定されていないので、コントローラを動作させるためにはまずこの機能を使用します。

● ライセンス設定

自律型コントローラ FCN/FCJ 基本ソフトウェアおよびアプリケーションポートフォリオの実行ライセンスを登録するための機能です。

● CPU モジュール設定

CPU モジュールに関する属性を設定する機能です。制御ネットワーク二重化に関する属性などを設定できます。

● I/O モジュール設定

I/O モジュールに関する属性を設定する機能です。コントローラに装着されている I/O モジュール構成を取得し I/O モジュールごとに属性を設定できます。

● RAS 情報の表示

FCN/FCJ の RAS（Reliability Availability Serviceability、以降 RAS と記述）情報を表示する機能です。

● ループチェックツール

ループチェックツールは、FCN/FCJ の I/O の配線チェック（ループチェック）を支援するツールです。I/O の現在値の確認や、出力値を直接設定できます。

● その他

・ APC 指令送信
FCN の CPU モジュールが二重化されているときに、待機側 CPU を有効にする（APC 指令）機能です。

B4.1.2 ロジックデザイナ

FCN/FCJ の制御ロジック作成ツール「ロジックデザイナ」は、以下の機能を備えています。

- プログラミング言語として、国際標準規格 IEC 61131-3 に準拠した 5 種類の言語をサポート。用途や好みに応じて使い分け、アプリケーションプログラムを構築できます。
- 制御ロジックを部品化（カプセル化）し、階層構造で制御アプリケーションを構築できるため、設計からデバッグまでエンジニアリング効率の向上が図れます。
- 強力なデバッグ機能を装備しています。

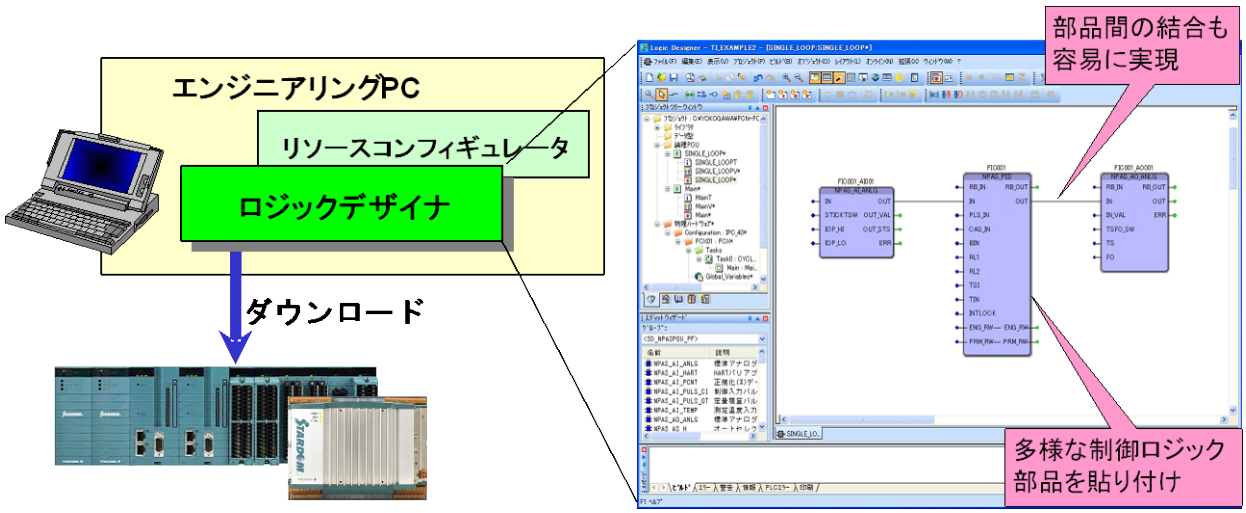


図　ロジックデザイナ

● POU とは

POU とは、Program Organization Unit（プログラム構成単位）の略で、制御アプリケーションを構成するプログラムコードをもつ、独立したソフトウェア単位です。

■ グローバルスタンダード言語 IEC 61131-3

FCN/FCJ の制御アプリケーションは、IEC 61131-3 準拠の 5 種類の言語を好みに応じて使い分けたり、用途に応じて組み合わせて使用できます。また、ループ制御とシーケンス制御が同一の開発環境で構築できるので、さらにエンジニアリングの効率化が図れます。

●5 種類の言語をサポート

- FBD（Function Block Diagram：機能ブロック図）
  各種機能がブロック化されており、そのブロック間を信号線で接続すればデータ処理を実現できます。データの流れに従って構築できるので、アナログ信号の連続制御に最適です。
- LD（Ladder Diagram：ラダー図）
  PLC のプログラミング言語として最も広く使われている言語で、リレーシンボルからきた接点やコイルなど構成要素を用いて、グラフィカルに論理演算を記述する言語です。
- SFC（Sequential Function Chart：シーケンスフローチャート）
  工程進捗型のアプリケーションを構築するのに便利な言語です。順序制御が明示的に記述できるメリットがあります。順序制御は SFC で記述する、異常処理など非順序制御は LD で記述する、などの組み合わせで簡潔なシーケンスプログラムが構築できます。
- IL（Instruction List：命令リスト）
  ニモニックを規格化した言語です。1 行に一つの演算子とオペランドで構成されているので表現に曖昧さがない反面、複雑な論理関係の記述は困難です。
- ST（Structured Text：構造化テキスト）
  高級言語である PASCAL 言語に似たテキスト言語で、処理が多岐に分かれるアプリケーションを IF-THEN-ELSE 文で記述でき、複雑な演算式も容易に記述できるなどのメリットがあります。

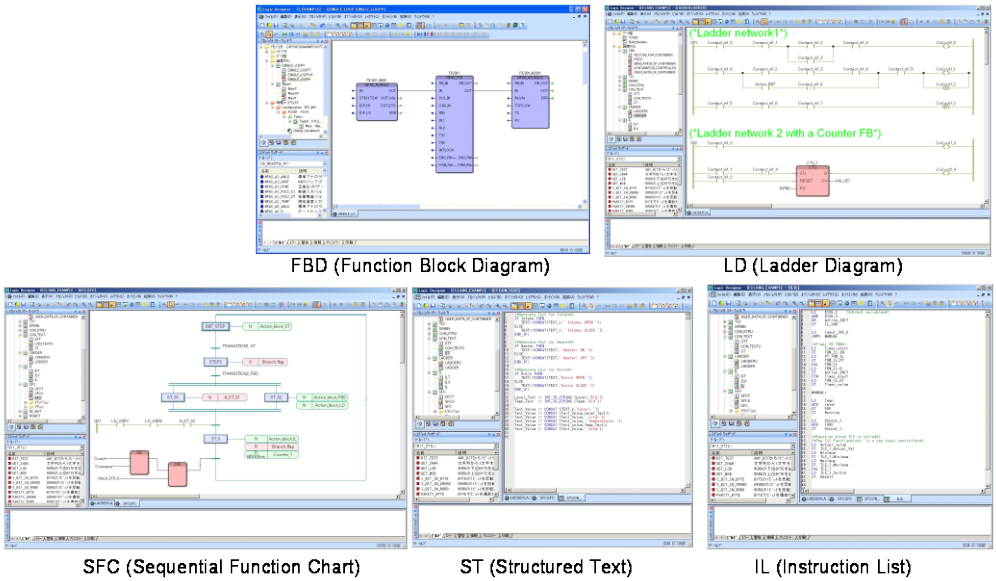


図　グローバルスタンダード言語 IEC 61131-3

■ 制御ロジックの部品化と再利用

FCN/FCJ は、制御アプリケーションノウハウをユーザレベルで簡単に部品化できます。
さらに、作成した部品をライブラリ化すれば、アプリケーションノウハウのカプセルが作成でき、再利用が可能となります。さらにアプリケーションを階層的に設計できます。
デバッグも含めたエンジニアリングの効率化に加え、プログラムの再利用による増改造も強力にサポートします。

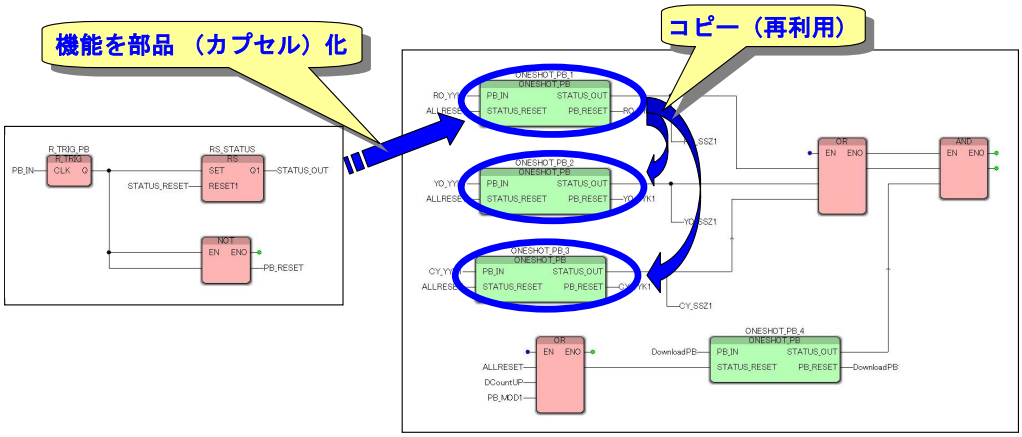


図　制御ロジックの部品化例

■ ネットワークテンプレートについて

“ネットワーク”とは、制御ロジックの結線を意味します。すなわち、ネットワークテンプレートとは、「制御ロジック（回路図）の雛形」です。雛形コードをテンプレート化して再利用できます。ネットワークテンプレートを活用すれば、簡単に制御ロジックを構築できます。
基本的な制御ループのテンプレートが「YOKOGAWA PLUS」（https://plus.yokogawa.co.jp/Membership/Top.do）の STARDOM 会員サイトにて公開されています。テンプレートをダウンロードして、所定のフォルダにコピーした上で利用してください。

参照

詳細については、テンプレートと一緒に公開されている説明資料を参照してください。

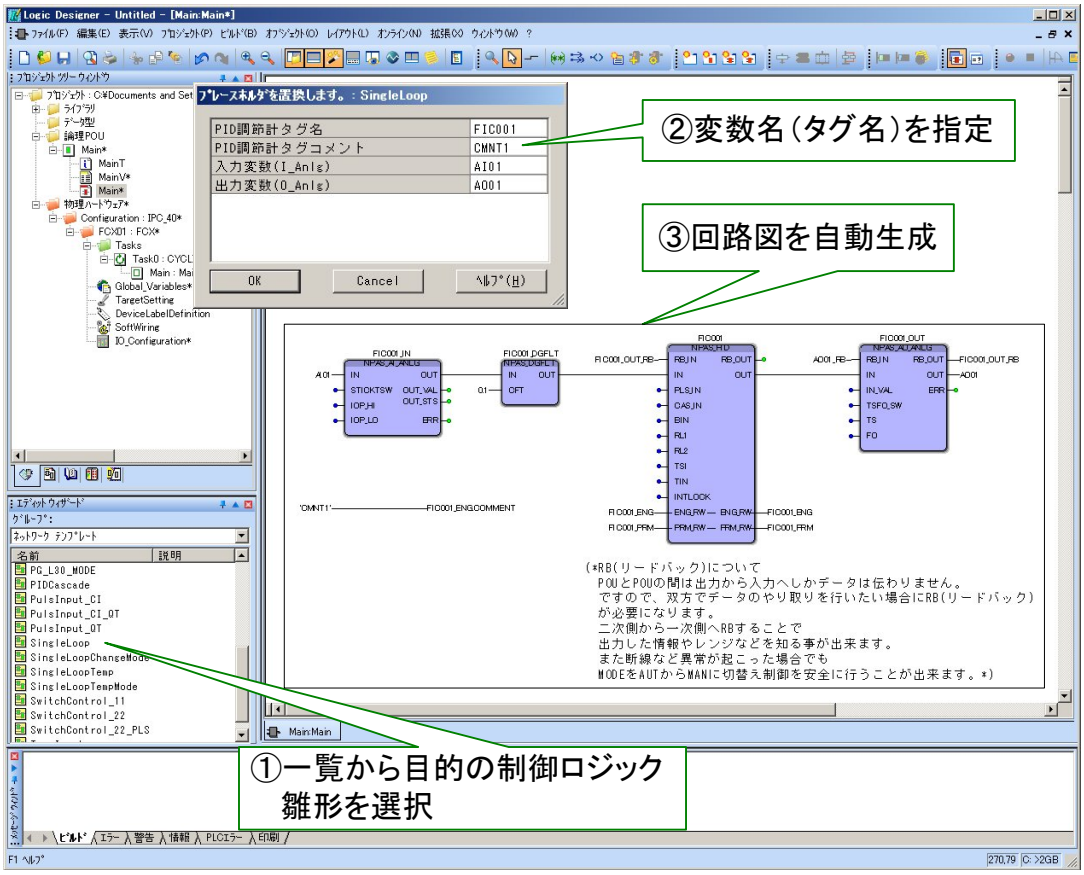


図　　ネットワークテンプレートの例

ネットワークテンプレートは、R2.01.01 以降のロジックデザイナで使用できます。

■ デバッグ支援機能

制御ロジック作成ツール「ロジックデザイナ」には、作成した制御ロジックのデバッグやメンテナンスを強力にサポートする各種機能も備わっているので、デバッグのためにプログラムを別途作成する必要がありません。

- ソフトワイヤリング機能
  I/O モジュールへの結線定義をソフト的に配線し、正弦波などの模擬信号を入力すれば、外部より信号を入力せずにデバッグできます。
- オンラインレイアウトとオンライン値表示
  プログラムを表示しながらプログラム中の変数の値を表示／変更できます。
- ブレークポイント設定
  プログラム中に設定したブレークポイントでプログラム実行が停止します。その後はステップ実行も可能です。
- ロジックアナライザ機能
  指定した時間間隔で変数の値を記録し、グラフィカルに表示します。
- ウォッチウィンドウ
  ウォッチウィンドウに変数を登録すれば変数の値をモニタリングできます。

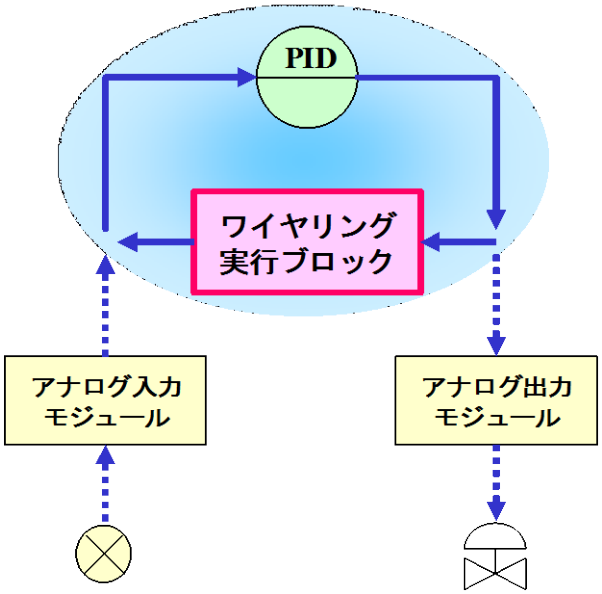


図　ソフトワイヤリング機能

■ FCN/FCJ シミュレータ

FCN/FCJ シミュレータは、1 台の FCN/FCJ で動作する制御アプリケーションを汎用 PC 上でシミュレーションするためのソフトウェアです。
この FCN/FCJ シミュレータにより、制御アプリケーションのデバッグなどが容易になり、開発効率を向上できます。

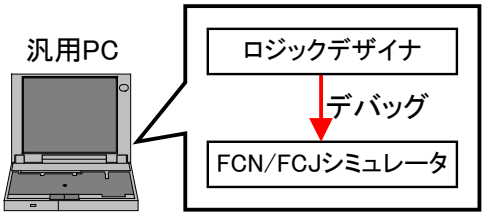


図　FCN/FCJ シミュレータ（例 1）

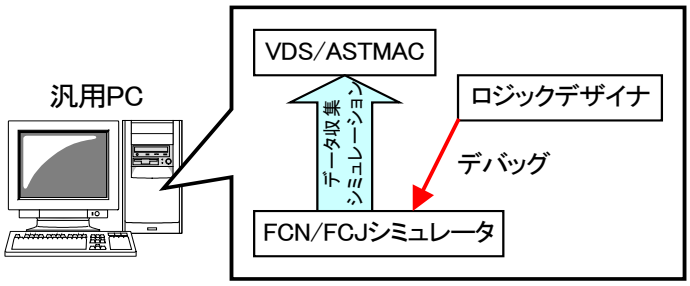


図　FCN/FCJ シミュレータ（例 2）

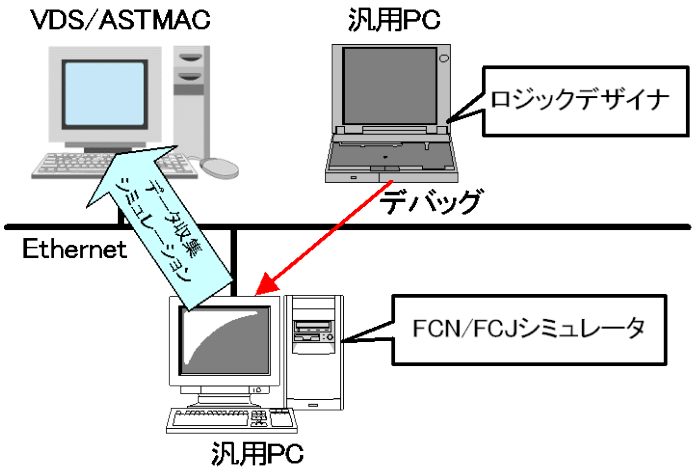


図　FCN/FCJ シミュレータ（例 3）

● シミュレーション

FCN/FCJ シミュレータ機能により、以下に示すデバッグが可能です。

- ロジックデザイナで作成した FCN/FCJ の制御アプリケーションの FCN/FCJ シミュレータによるデバッグ
- VDS/ASTMAC の FCN/FCJ シミュレータへのデータアクセスによるデバッグ
- FCN/FCJ 上で制御アプリケーションと連動する Java アプリケーションの汎用 PC 上におけるデバッグ

FCN/FCJ シミュレータは、ロジックデザイナや VDS/ASTMAC と共に 1 台の汎用 PC 上において動作させられます。また、別の汎用 PC 上においても動作させられます。
1 台の汎用 PC 上で動作可能な FCN/FCJ シミュレータは、1 台分だけです。

● シミュレータで利用できない機能

FCN/FCJ シミュレータでは、下記機能は利用できません。

- FCN/FCJ の I/O 機能
  I/O は常に切り離し状態となります。
- Web ブラウザによるコンフィギュレーション機能
- CPU の二重化に関連する機能
- ネットワークの二重化に関連する機能
- FCN/FCJ 間通信または各種通信機能
- リテインデータのハードバックアップ

● シミュレータで利用できない POU および APPF

FCN/FCJ シミュレータで、POU や APPF（FCN/FCJ アプリケーションポートフォリオ）を用いても外部機器とは通信できません。

FCN/FCJ シミュレータ上でエラーとなる標準 POU（通信を伴う POU すべて）

- FCN/FCJ 間通信ファンクションブロック
- FA-M3 エミュレーション（シリアル）ファンクションブロック
- Ethernet 通信ファンクションブロック
- シリアル通信ファンクションブロック

FCN/FCJ シミュレータ上でエラーとなる主な APPF

- FA-M3 通信ポートフォリオ
- MELSEC 通信ポートフォリオ
- SYSMAC 通信ポートフォリオ
- Modbus 通信ポートフォリオ
- DNP3 通信ポートフォリオ
- 温調計通信ポートフォリオ

● その他

- Java アプリケーションの実行のためには、Java の実行環境を汎用 PC 上に準備する必要があります。FCN/FCJ Java アプリケーション開発キット R1.30.01 以上が必要です。
- FCN/FCJ シミュレータの処理速度は、汎用 PC の性能により異なります。FCN/FCJ シミュレータは、FCN/FCJ より高速に処理を実行する場合があります。制御アプリケーションの実行速度は、必ず FCN/FCJ で確かめてください。

B4.1.3 ファンクション／ファンクションブロック

以下にロジックデザイナが標準で用意しているファンクション、ファンクションブロックの一覧を示します。

参照

- ロジックデザイナが標準で用意しているファンクション、ファンクションブロック各機能の詳細については、「FCN/FCJ ファンクション・ファンクションブロック説明書」（TI 34P02K32-01）を参照してください。
- 制御アプリケーションの構成要素であるファンクション／ファンクションブロックの位置づけについては B4.3.1 の「■論理 POU」を参照してください。

■ IEC 61131-3 のファンクション・ファンクションブロック

IEC 61131-3 で用意されるファンクションとファンクションブロックの一覧を示します。

表　型変換ファンクション

ファンクション	説明
*_TO_STRING	各型から STRING への型変換、*部分は、BYTE、DWORD、INT、DINT、REAL、LREAL、SINT、TIME、USINT、UINT、UDINT、WORD
B_BCD_TO_*	BYTE 型 BCD コードから各型への型変換、*部分は、DINT、INT、SINT
BCD_TO_DINT	DWORD 型 BCD コードから DINT への型変換
BOOL_TO_*	BOOL から各型への型変換、*部分は、BYTE、WORD、DWORD、SINT、INT、DINT、USINT、UINT、UDINT、REAL
BYTE_TO_*	BYTE から各型への型変換、*部分は、BOOL、WORD、DWORD、SINT、INT、DINT、USINT、UNIT、UDINT、REAL
D_BCD_TO_*	DWORD 型の BCD コードから各型への型変換、*部分は、SINT、INT、DINT
DINT_TO_*	DINT から各型への型変換、*部分は、BCD、B_BCD、W_BCD、D_BCD、BOOL、BYTE、WORD、DWORD、SINT、INT、REAL、LREAL、TIME、USINT、UNIT、UDINT
DWORD_TO_*	DWORD から各型への型変換、*部分は、BOOL、BYTE、WORD、SINT、INT、DINT、REAL、USINT、UINT、UDINT
INT_TO_*	INT から各型への型変換、*部分は、B_BCD、W_BCD、D_BCD、BOOL、BYTE、WORD、DWORD、SINT、DINT、REAL、USINT、UINT、UDINT
LREAL_TO_*	REAL から各型への型変換、*部分は、DINT、REAL、UDINT
REAL_TO_*	REAL から各型への型変換、*部分は、BOOL、BYTE、WORD、DWORD、SINT、INT、DINT、LREAL、USINT、UINT、UDINT
SINT_TO_*	SINT から各型への型変換、*部分は、B_BCD、W_BCD、D_BCD、BOOL、BYTE、WORD、DWORD、INT、DINT、REAL、USINT、UINT、UDINT
STRING_TO_*	STRING から各型への型変換、*部分は、BYTE、WORD、DWORD、SINT、INT、DINT、REAL、TIME、USINT、UINT、UDINT
TIME_TO_DINT	TIME から DINT への型変換
TRUNC_*	REAL 型データを小数点以下切り捨てて、各型へ型変換、*部分は、SINT、INT、DINT
USINT_TO_*	USINT から各型への型変換、*部分は、BOOL、BYTE、WORD、DWORD、SINT、INT、DINT、UINT、UDINT、REAL
UINT_TO_*	UINT から各型への型変換、*部分は、BOOL、BYTE、WORD、DWORD、SINT、INT、DINT、USINT、UDINT、REAL
UDINT_TO_*	UDINT から各型への型変換、*部分は、BOOL、BYTE、WORD、DWORD、SINT、INT、DINT、USINT、UINT、REAL、LREAL
W_BCD_TO_*	WORD 型 BCD コードから各型への型変換、*部分は、SINT、INT、DINT
WORD_TO_*	WORD から各型への型変換、*部分は、BOOL、BYTE、DWORD、SINT、INT、DINT、REAL、USINT、UINT、UDINT

表　数値ファンクション

ファンクション	説明
ABS	絶対値
ACOS	アークコサイン
ASIN	アークサイン
ATAN	アークタンジェント
COS	コサイン
EXP	指数
LN	自然対数
LOG	常用対数
SIN	サイン
SQRT	平方根
TAN	タンジェント

表　算術ファンクション

ファンクション	説明
ADD	ANY_NUM 型の加算
ADD_T_T	TIME 型の加算
DIV	ANY_NUM 型の除算
DIV_T_AN	TIME 型の除算
EXPT	べき乗
MOD	剰余算
MOVE	転送
MUL	ANY_NUM 型の乗算
MUL_T_AN	TIME 型の乗算
NEG	2 の補数
SUB	ANY_NUM 型の減算
SUB_T_T	TIME 型の減算

表　ブールビット演算ファンクション

ファンクション	説明
AND	AND 演算
NOT	補数演算
OR	OR 演算
XOR	XOR 演算

表　ビット文字列ファンクション

ファンクション	説明
ROL	左ローテーション
ROL_*	型指定左ローテーション、*部分は BYTE、WORD、DWORD
ROR	右ローテーション
ROR_*	型指定右ローテーション、*部分は BYTE、WORD、DWORD
SHL	左シフト
SHL_*	型指定左シフト、*部分は BYTE、WORD、DWORD
SHR	右シフト
SHR_*	型指定右シフト、*部分は BYTE、WORD、DWORD

表　選択ファンクション

ファンクション	説明
LIMIT	限定値
LIMIT_*	型指定限定値、*部分は、INT、DINT、SINT、REAL、STRING
MAX	最大値
MAX_*	型指定最大値、*部分は、INT、DINT、SINT、REAL、STRING
MIN	最小値
MIN_*	型指定最小値、*部分は、INT、DINT、SINT、REAL、STRING
SEL	バイナリ選択
SEL_*	型指定バイナリ選択、*部分は、BOOL、BYTE、DINT、INT、SINT、REAL、STRING、WORD、DWORD、TIME

表　比較ファンクション

ファンクション	簡単な説明
EQ	比較:=
GE	比較:>=
GT	比較:>
LE	比較:<=
LT	比較:<
NE	比較:<>

表　文字列ファンクション

ファンクション	説明
CONCAT	拡張可能な連結
DELETE	文字列の削除
EQ_STRING	比較:=（STRING に対して）
FIND	文字の検索
GE_STRING	比較:>=（STRING に対して）
GT_STRING	比較:>（STRING に対して）
INSERT	文字の挿入
LE_STRING	比較:<=（STRING に対して）
LEFT	左端からの文字
LEN	文字列の長さ
LT_STRING	比較:<（STRING に対して）
MID	中間からの文字
NE_STRING	比較:<>（STRING に対して）
REPLACE	文字の置き換え
RIGHT	右端からの文字

表　バイステーブルファンクションブロック

ファンクションブロック	説明
SR	優先権のセット
RS	優先権のリセット

表　エッジ検出型ファンクションブロック

ファンクションブロック	説明
F_TRIG	立ち下がりエッジ検出
R_TRIG	立ち上がりエッジ検出

表　カウンタファンクションブロック

ファンクションブロック	説明
CTU	アップカウンタ
CTD	ダウンカウンタ
CTUD	アップダウンカウンタ

表　タイマファンクションブロック

ファンクションブロック	説明
TP	パルス
TON	オンディレイタイマ
TOF	オフディレイタイマ

■ ビット操作ファンクション

ビット操作ファンクションの一覧を以下に示します。本ファンクションは、ライブラリ"Bit_Util"で提供されます。

表　ビット操作ファンクション

ファンクション	説明
BIT_TEST	ビット文字列の 1 ビットの値の読み出し
GET_CHAR	文字列からの 1 文字の抽出
GET_LSB	ビット文字列の下位バイトの値の読み出し
GET_MSB	ビット文字列の上位バイトの値の読み出し
I_BIT_IN_*	ビット文字列の 1 ビットの反転、*部分は、BYTE、WORD、DWORD
PARITY_*	セットビット数が偶数か奇数かのチェック、*部分は、BYTE、WORD、DWORD
R_BIT_IN_*	ビット文字列の 1 ビットをリセット、*部分は、BYTE、WORD、DWORD
S_BIT_IN_*	ビット文字列の 1 ビットをセット、*部分は、BYTE、WORD、DWORD
SET_LSB	ビット文字列の下位バイトに値を書き込み
SET_MSB	ビット文字列の上位バイトに値を書き込み
STRING_TO_BUFFER	文字列の文字をバッファにコピー
SWAP	ビット文字列の上位バイトと下位バイトを入れ替え

■ FCN/FCJ 基本ファンクションブロック

FCN/FCJ 基本ファンクションブロックの一覧を以下に示します。

表　FCN/FCJ 基本ファンクションブロック

ファンクションブロック	説明
PAS_GET_POUINSTN	POU のインスタンス名を取得
PAS_GET_TASKDELAY	POU を動作させるタスクの遅延時間を取得
PAS_GET_TASKSCAN	POU を動作させるタスクのスキャン周期を取得
PAS_GETTIME	ローカルタイムを取得
PAS_GETTIME2	ローカルタイムをミリ単位で取得
PAS_SETTIME	接点入力等、外部の事象により時刻（ローカルタイム）を設定
PAS_MSG_UPRCALM	ユーザが指定した文字列をプロセスアラームとして発行
PAS_MSG_USRALM	ユーザが指定した文字列をアラームとして送信
PAS_MSG_USREVT	ユーザが指定した文字列をイベントとして送信
PAS_START_ACT	ワインドアップカウンタのカウントダウンやアプリケーション実行可能状態フラグの操作
PAS_WUP_CHECK	ワインドアップの状態をチェック
PAS_WUP_CT_INIT	ワインドアップカウンタの初期値を取得
PAS_UNIT_TO_BYTE	工業単位文字列を STRING 型から BYTE 型へ変換
PAS_UNIT_TO_STRING	工業単位文字列を BYTE 型から STRING 型へ変換
PAS_OPR_PRM_DWORD	DWORD 型データの各ビットに対する操作
PAS_IS_OLDL_DONE	オンラインダウンロードの実行を検出
PAS_GET_TASKDELAY	オンラインダウンロードの実行によるユーザタスクの遅延時間を取得
SD_FIELD_LANCTRL	Ethernet 動作状態の切替を行うための POU
SD_FIELD_LANGETSTS	Ethernet 動作状態の取得を行うための POU
SD_FIELD_ACCS_CTRL	アクセスコントロールの変更

注：
- FCN/FCJ 基本ファンクションブロックのライブラリ名は、"SD_FIELD_PF"です。
- FCN/FCJ 基本ファンクションブロックは、ファンクションブロック名が「PAS_***」となっていますが、PAS ポートフォリオのライセンスは不要です。

■ FCN/FCJ 間通信ファンクションブロック

FCN/FCJ 間通信ファンクションブロックは、FCN/FCJ 間通信で使用するファンクションブロックです。FCN/FCJ 間通信ファンクションブロックの一覧を以下に示します。

表　FCN/FCJ 間通信ファンクションブロック

ファンクションブロック	説明	備考
CONNECT	通信相手とのコネクト処理	
USEND_1V	通信相手にデータを送信（Unconfirmed）	1 回の要求で 1 個のデータを送信
USEND_5V	通信相手にデータを送信（Unconfirmed）	1 回の要求で 5 個のデータを送信
USEND_10V	通信相手にデータを送信（Unconfirmed）	1 回の要求で 10 個のデータを送信
URECV_1V	通信相手からのデータを受信（Unconfirmed）	1 回の要求で 1 個のデータを受信
URECV_5V	通信相手からのデータを受信（Unconfirmed）	1 回の要求で 5 個のデータを受信
URECV_10V	通信相手からのデータを受信（Unconfirmed）	1 回の要求で 10 個のデータを受信
WRITE_1V	通信相手にデータを送信（Confirmed）	1 回の要求で 1 個のデータを送信
WRITE_5V	通信相手にデータを送信（Confirmed）	1 回の要求で 5 個のデータを送信
WRITE_10V	通信相手にデータを送信（Confirmed）	1 回の要求で 10 個のデータを送信
READ_1V	通信相手からのデータを受信（Confirmed）	1 回の要求で 1 個のデータを受信
READ_5V	通信相手からのデータを受信（Confirmed）	1 回の要求で 5 個のデータを受信
READ_10V	通信相手からのデータを受信（Confirmed）	1 回の要求で 10 個のデータを受信

注：　FCN/FCJ 間通信ファンクションブロックのライブラリ名は、"SD_FCXCOM_LIB"です。

■ FA-M3 エミュレーション（シリアル通信）ファンクションブロック

FA-M3 エミュレーション（シリアル通信）ファンクションブロックは、FCN/FCJ の COM ポートと RS-232-C 経由で接続された上位コンピュータ（または表示器）から送出される、FA-M3 用のコマンドをエミュレートするファンクションブロックです。FA-M3 エミュレーション（シリアル通信）ファンクションブロックの一覧を以下に示します。

表　FA-M3 エミュレーション（シリアル通信）ファンクションブロック

ファンクションブロック	説明
SD_CFAM3R_OPEN	FA-M3 エミュレートタスクを起動し、RS-232-C の通信を開始

注：　FA-M3 エミュレーション（シリアル通信）ファンクションブロックのライブラリ名は、"SD_FCXPLCR_LIB"および"SD_CFAM3R_PF"です。

■ 通信ユーティリティファンクションブロック

通信ユーティリティファンクションブロックは、各種通信ライブラリで使用する補助機能を提供します。通信ユーティリティファンクションブロックの一覧を以下に示します。本ファンクションブロックは、ライブラリ"SD_CUTIL_PF"で提供されます。

表　通信ユーティリティファンクションブロック

ファンクションブロック	説明
SD_CUTIL_TRIG	通信要求トリガ生成
SD_CUTIL_W2DW	2 つの WORD 型データから DWORD 型へのデータ変換
SD_CUTIL_W2R	2 つの WORD 型データから REAL 型へのデータ変換
SD_CUTIL_DW2W	DWORD 型から 2 つの WORD 型データへのデータ変換
SD_CUTIL_DW2R	DWORD 型から REAL 型へのデータ変換
SD_CUTIL_R2W	REAL 型から 2 つの WORD 型データへのデータ変換
SD_CUTIL_R2DW	REAL 型から DWORD 型へのデータ変換
SD_CUTIL_WORDSWAP	WORD 型データのエンディアン変換
SD_CUTIL_DWORDSWAP	DWORD 型データのエンディアン変換

注：　通信ユーティリティファンクションブロックのライブラリ名は、" SD_CUTIL_PF" です。

■ Ethernet 通信ファンクションブロック

Ethernet 通信ファンクションブロックは、FCN/FCJ と Ethernet で接続された機器との通信機能を提供します。Ethernet 通信ファンクションブロックの一覧を以下に示します。

表　Ethernet 通信ファンクションブロック

ファンクションブロック	説明
SD_FCXPE_OPEN	通信タスクの起動
SD_FCXPE_SENDRECV	送受信要求

注：　Ethernet 通信ファンクションブロックのライブラリ名は、“SD_FCXPLCE_PF”です。

■ シリアル通信ファンクションブロック

シリアル通信ファンクションブロックは、FCN/FCJ と RS 通信で接続された機器との通信機能を提供します。シリアル通信ファンクションブロックの一覧を以下に示します。

表　シリアル通信ファンクションブロック

ファンクションブロック	説明
SD_FCXPM_OPEN	通信タスク起動処理
SD_FCXPM_OPEN_EX	拡張通信タスク起動処理
SD_FCXPM_SENDRECV	送受信処理
SD_FCXPM_SEND	送信処理
SD_FCXPM_RECV	受信処理
SD_FCXPM_GETCH	チャネル番号取得処理
SD_FCXPS_OPEN	通信タスク起動処理
SD_FCXPS_SENDRECV	送受信処理（旧版ファンクションブロック）
SD_FCXPS_SEND	送信処理（旧版ファンクションブロック）
SD_FCXPS_RECV	受信処理（旧版ファンクションブロック）

注：　シリアル通信ファンクションブロックのライブラリ名は、“SD_FCXPLCS_PF”です。

B4.1.4 アプリケーションポートフォリオ（APPF）

当社が長年培ったアプリケーションノウハウをライブラリ化した「アプリケーションポートフォリオ（APPF）」をオプションで用意しています。PID 調節計、手動操作器といった基本的なカプセルから、プロセスや装置に特化したカプセルまで必要なものを選択し、対象に合わせてモディファイすれば高度な制御アプリケーションを簡単かつ確実に構築できます。また、標準部品を利用すればデバッグやメンテナンスの効率化も図れます。

- **PAS ポートフォリオ**
  PID 調節計やスイッチ計器などプロセス制御用部品
- **通信ポートフォリオ**
  PLC（FA-M3、MELSEC、SYSMAC）、Modbus 対応機器、DNP3 対応機器や横河製デジタル指示調節計との通信機能
- **SAMA ポートフォリオ**
  電力 SAMA 仕様に準拠した部品
- **AP 特化ポートフォリオ**
  アプリケーションに特化した機能を部品化（ボイラ制御ポートフォリオ、ボイラ補機制御ポートフォリオ）

参照

- PAS ポートフォリオについては、「Part-D NPAS POU 編」を参照してください。
- 通信ポートフォリオ、SAMA ポートフォリオ、AP 特化ポートフォリオについては、本書「F3. GS 集：APPF 関連」「FCN/FCJ アプリケーションポートフォリオ」（GS 34P02P20-01）を参照してください。
- 通信ポートフォリオで扱えるデバイスについては、「通信ポートフォリオで利用可能なデバイス一覧」（TI 34P02P21-01）を参照してください。
- Modbus 通信については、本書「F3. GS 集：APPF 関連」「Modbus 通信ポートフォリオ」（GS 34P02P21-01）を参照してください。
- DNP3 通信については、本書「F3. GS 集：APPF 関連」「DNP3 通信ポートフォリオ」（GS 34P02P22-01）を参照してください。

B4.1.5 FCN/FCJ Java アプリケーション開発

FCN/FCJ の Java アプリケーションで実現する情報発信機能は、情報発信パッケージ InfoWell でそのほとんどが、プログラミングレスで実現できます。
情報発信パッケージ InfoWell で実現できない機能は、Java アプリケーションで開発する必要があります。

参照

情報発信パッケージ InfoWell については、「B2.3.1 情報発信パッケージ InfoWell」を参照してください。

■ ユーザによる Java アプリケーションの開発

Java アプリケーションは、 Oracle 社製 JDK（JDK 1.1.8）で開発しますが、開発する上で必要となる以下の機能は、FCN/FCJ Java アプリケーション開発キットで提供されます。

- JADE クラスライブラリ
  Java アプリケーションのマルチタスク実行を管理するためのクラスライブラリ
- JEROS クラスライブラリ
  システムを管理するためのクラスライブラリ
- 制御データアクセスクラスライブラリ
  制御機能のデータにアクセスするためのクラスライブラリ

補足

- 上記各クラスライブラリは、開発する PC 上にインストールし使用します。FCN/FCJ 上には標準で装備されています。
- Java 標準クラスライブラリは、 Oracle 社製 JDK（Java Development Kit）で提供されます。

- Duolet モニタ
  Windows マシンから、ネットワーク経由で FCN/FCJ 上で動作している Java アプリケーション（Duolet）をモニタ・デバッグする機能
- JEROS エミュレータ
  Windows 上に JEROS システムサービスをエミュレートし、Java アプリケーションをテストする機能（JEROS エミュレータは、制御データアクセスクラスには対応していません。制御データにアクセスする Java アプリケーションは FCN/FCJ 上でテストしてください。）

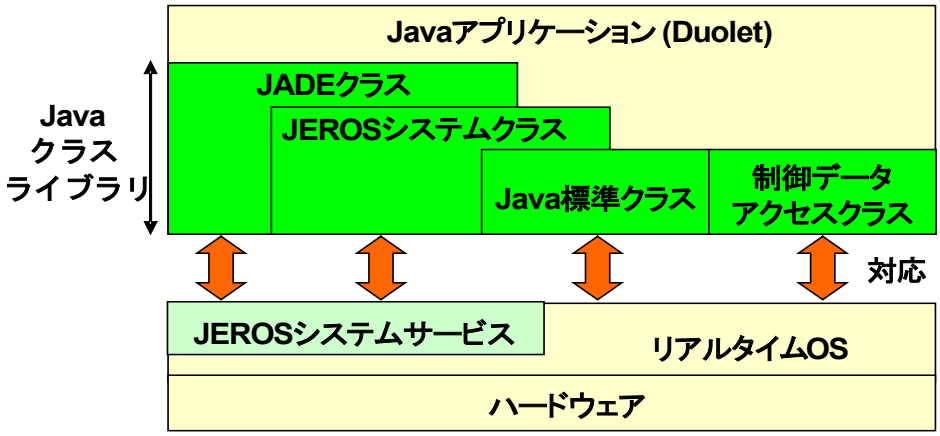


図　FCN/FCJ Java 機能の構成

● Webmetry 基本ライブラリポートフォリオ

Webmetry 基本ライブラリポートフォリオは、FCN/FCJ の Java 機能上で動作する Web 監視アプリケーションの作成を支援するアプリケーションポートフォリオです。
瞬時値バーグラフ表示、瞬時値デジタル表示、トレンド表示といった Web ブラウザで監視が可能な画面を最小限のコーディングで作成できます。

補足

Webmetry 基本ライブラリポートフォリオで Web 監視アプリケーションを作成するには、FCN/FCJ Java アプリケーション開発キットが必要です。また、作成した Web 監視アプリケーションを実行する各 FCN/FCJ には本ライセンスが登録されている必要があります。

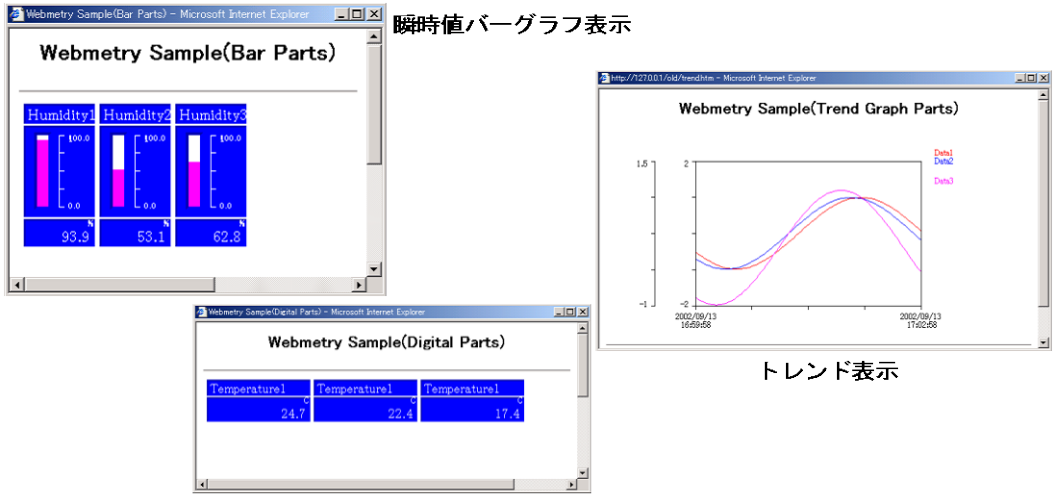


図　Webmetry 基本ライブラリポートフォリオによる画面表示例

B4.1.6 Web ブラウザによる FCN/FCJ の設定

FCN/FCJ に対して詳細情報を設定および操作する場合には、Web ブラウザを使用します。本章では、Web ブラウザを使った FCN/FCJ の高度な設定について説明します。
FCN/FCJ のメンテナンスホームページは、下図のような構成になっています。

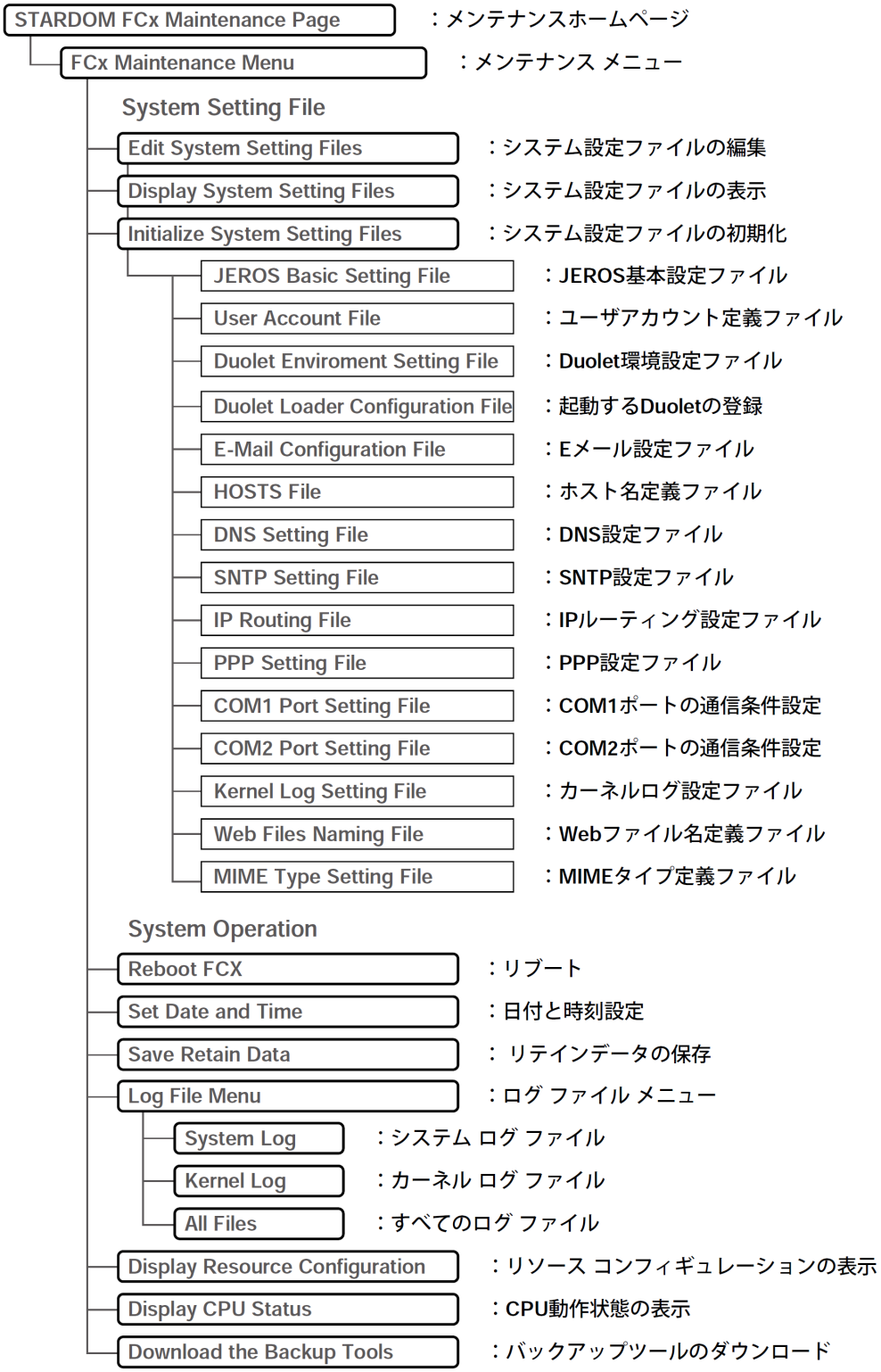


図　FCN/FCJ メンテナンスホームページの構成

■ メンテナンスホームページ

Web ブラウザが FCN/FCJ と接続されると、以下に示すメンテナンスホームページが表示されます。このページでは、以下に示す FCN/FCJ の各種属性が表示されます。

- ホスト名
- IP アドレス
- サブネットマスク
- MAC アドレス（PORT1）
- FCN/FCJ のタイプ
- ハードウェアシリアル番号
- システムカードシリアル番号
- OS のバージョン
- ブート ROM のバージョン
- JEROS バージョン
- 実装メモリ容量
- SRAM 容量
- FCN/FCJ システムカード容量
- FCN/FCJ システムカード空き容量
- 日付と時刻
- 動作状態（Maintenance or Online）

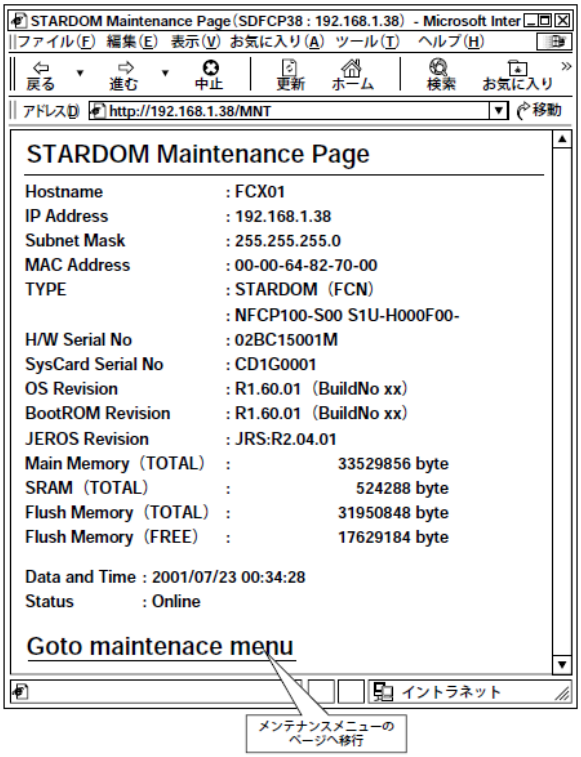


図　メンテナンスホームページ

参照

メンテナンスホームページの詳細については、各設定画面のヘルプを参照してください。

B4.2 FCN/FCJ システム構築手順

FCN/FCJ システムは、基本的に以下の手順で構築します。

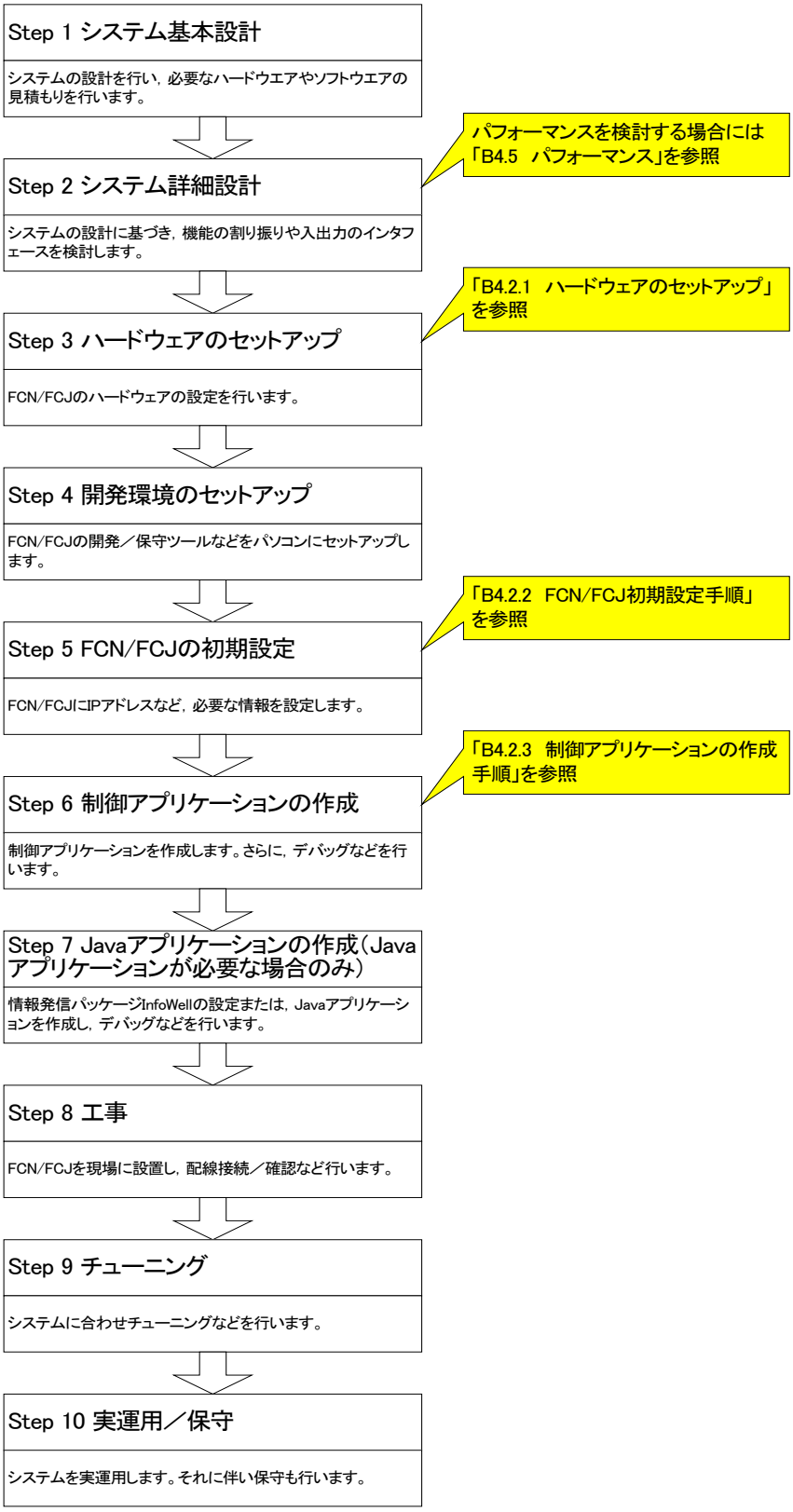


図　FCN/FCJ システム構築手順

B4.2.1 ハードウェアのセットアップ

本項では、下記のモジュールに必要なハードウェア初期設定の方法を述べます。

- ベースモジュール NFBU200
  ユニット番号の設定、CPU モジュール台数の設定
- 電流入力モジュール NFAI141、NFAI143
  伝送器の種類により電源供給の有／無の設定
- 電流入出力モジュール NFAI841
  伝送器の種類により電源供給の有／無の設定

■ ベースモジュール NFBU200 の設定

ベースモジュールには、ハードウェア初期設定として、
- ユニット番号設定
- CPU モジュール台数設定（NFCP100 二重化の有無）

があります。

● ユニット番号設定（SW1）

P1 スロット横 SW1 を、使用するユニット番号に対応したシルク図に従って設定してください。（工場出荷時の設定：ユニット 1）

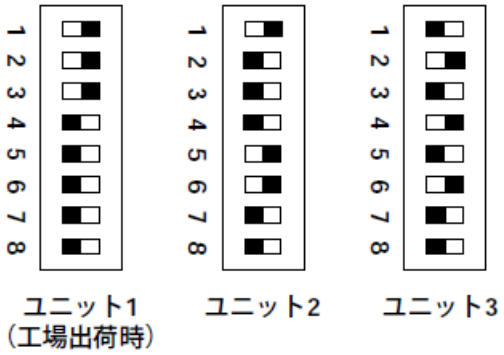


図　ユニット番号設定（SW1）

● CPU モジュール台数設定（SW2）

CPU を 2 台実装（制御ユニットで CPU モジュールを二重化）する場合には、Slot2 横 SW2 をシルク図［CPU×2］のように設定してください。
それ以外（増設ユニットまたはシングル CPU モジュールの制御ユニット）の場合には、Slot2 横 SW2 をシルク図［Other］のように設定してください。
（工場出荷時の設定：Other）

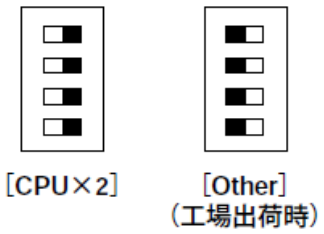


図　CPU モジュール台数設定（SW2）

重　要

- SW2 が［CPU×2］に設定されている場合には、必ず Slot3～Slot4 に CPU モジュールを実装してください。
- SW2 が［Other］に設定されている場合には、決して Slot3～Slot4 に CPU モジュールを実装しないでください。

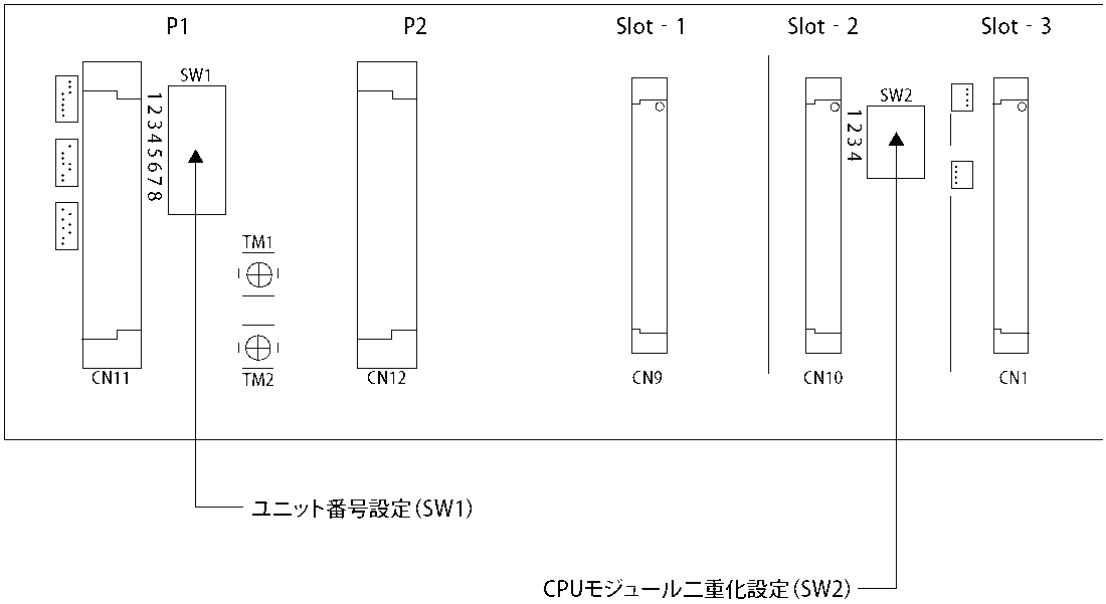


図　ベースモジュールの設定

B

■ NFAI141、NFAI143 の設定

電流入力モジュール（NFAI141、NFAI143）は、接続する伝送器への電源供給の有無を設定ピンで設定します。
2 線式伝送器と接続する場合、電源供給ありに設定します。
4 線式伝送器と接続する場合、電源供給なしに設定します。
設定ピンのピン番号（1～16）は、チャネル番号に対応しています。
下図のように設定ピンは、NFAI141、NFAI143 電流入力モジュール右側面にあります。

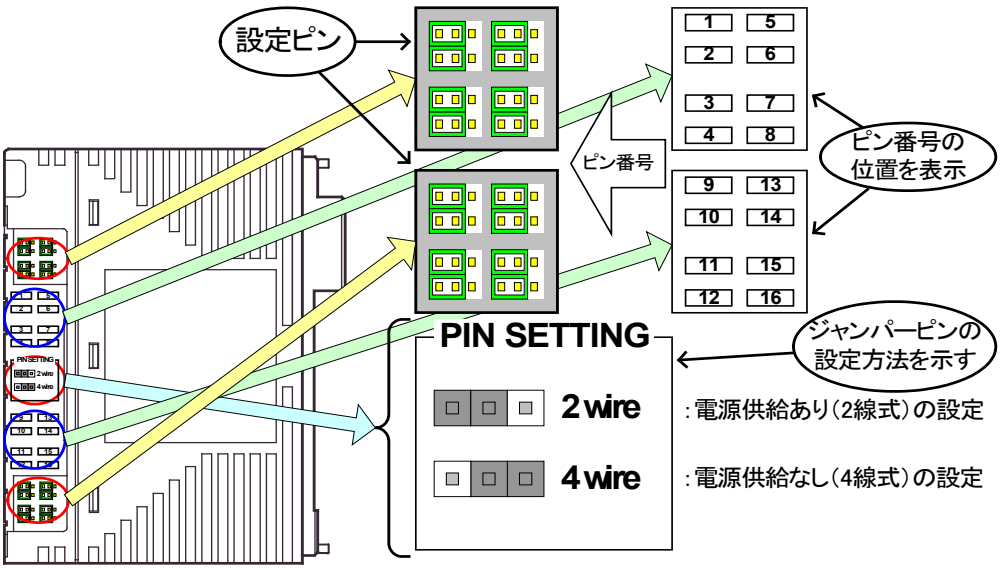


図　NFAI141、NFAI143 電流入力モジュールの設定

■ NFAI841 の設定

電流入出力モジュール（NFAI841）は、接続する伝送器への電源供給の有無を設定ピンで設定します。
2 線式伝送器と接続する場合、電源供給ありに設定します。
4 線式伝送器と接続する場合、電源供給なしに設定します。
設定ピンのピン番号（1～8）は、チャネル番号に対応しています。
下図のように設定ピンは、NFAI841 電流入出力モジュール右側面にあります。

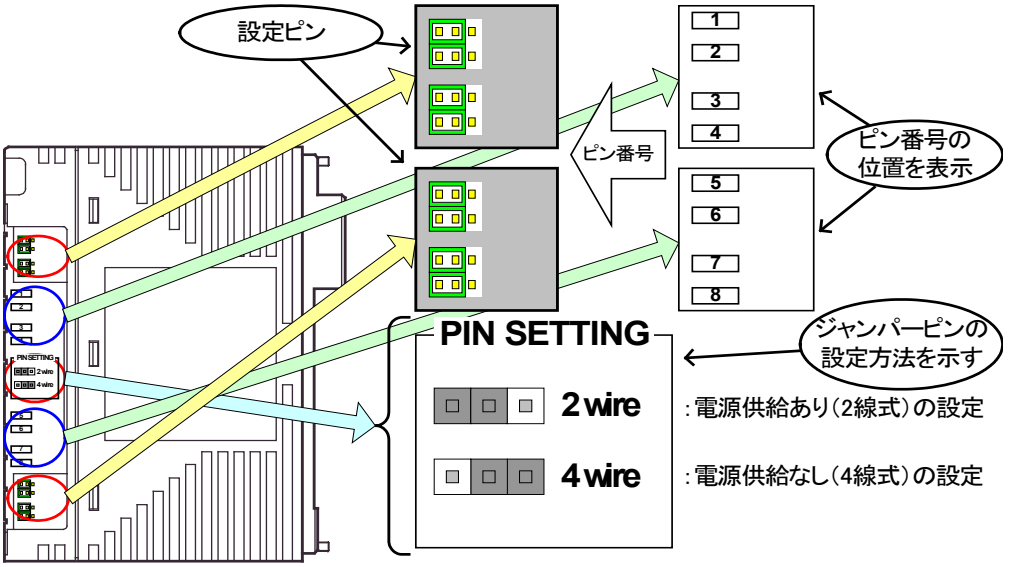


図　NFAI841 電流入出力モジュールの設定

補足

入力機能を使用時のみ、電源供給の有無の設定が可能です。

B4.2.2 FCN/FCJ 初期設定手順

システムの設計やハードウェアの初期設定を終了したら、FCN/FCJ の初期設定を開始します。
FCN/FCJ の初期設定手順は、以下のとおりです。

● 手順 1 ネットワークの設定

FCN/FCJ の工場出荷時には、ネットワーク関連情報（IP アドレスおよびネットマスク）が設定されていません。このため最初にリソースコンフィギュレータを用いてネットワークの属性を設定します。
設定する属性には、下記のものがあります。

- IP アドレス
- ネットマスク
- サブネットマスク
- ゲートウェイ
- タイムゾーン

重 要

初期状態では、タイムゾーンは、GMT になっています。使用する地域にあわせて、必ずタイムゾーンを修正してください。

● 手順 2 設定の反映

ネットワークの設定が終了したら、FCN/FCJ を再立ち上げてください。これにより FCN/FCJ は、設定された情報により立ち上がります。

● 手順 3 FCN/FCJ との接続

リソースコンフィギュレータから設定した IP アドレスを指定して、FCN/FCJ に接続します。これによりリソースコンフィギュレータは、FCN/FCJ の設定操作が可能になります。

● 手順 4 リソースコンフィギュレータによる基本設定

リソースコンフィギュレータにより、下記の情報を設定します。

- ライセンス
- CPU モジュール
- I/O モジュール

● 手順 5 設定情報のダウンロード

ライセンス、CPU モジュール、I/O モジュールの設定が完了したら、この情報を FCN/FCJ にダウンロードします。

● 手順 6 Web ブラウザによる接続

Web ブラウザから FCN/FCJ の IP アドレスを指定すれば、FCN/FCJ と接続できます。
Web ブラウザより設定する場合には、FCN/FCJ をメンテナンス状態にする必要があります。FCN/FCJ をメンテナンス状態でリブートしてください。

参照

- Web ブラウザと FCN/FCJ を接続する方法については、「STARDOM FCN/FCJ 解説書」（IM 34P02Q01-01）「B2.2 FCN/FCJ との接続」を参照してください。
- FCN/FCJ をメンテナンス状態でリブートする方法については、「STARDOM FCN/FCJ 解説書」（IM 34P02Q01-01）「B2.4.5 リブート」を参照してください。

● 手順 7 Web ブラウザからの設定

FCN/FCJ に対して Web ブラウザより各種の詳細を設定できます。
Web ブラウザがら設定可能な主な属性には、下記のものがあります。

- ホスト名
- IP アドレス
- DNS クライアントの起動
- タイムゾーン
- ユーザアカウント

参照

Web ブラウザで設定についての詳細は、「STARDOM FCN/FCJ 解説書」（IM 34P02Q01-01）「B2.4 メンテナンスメニュー」を参照してください。

● 手順 8 オンライン状態で再起動

FCN/FCJ に対して上記を設定したら、オンライン状態で再起動してください。

B4.2.3 制御アプリケーションの作成手順

FCN/FCJ の制御アプリケーションは、ロジックデザイナで作成します。初期設定の手順は、以下のとおりです。これらはリソースコンフィギュレータで設定します。

参照

制御アプリケーションの詳細な作成方法については、ロジックデザイナのヘルプ「プログラミングガイド」を参照してください。

● 手順 1 制御アプリケーション開発環境の準備

開発環境に必要な機材などを用意します。主なものとしては、以下のものがあります。

・汎用 PC
・ソフトウェア媒体（DVD-ROM）
・ロジックデザイナのライセンス
・FCN/FCJ
・FCN/FCJ と汎用 PC を接続するネットワーク環境

上記の環境が揃ったら、汎用 PC にロジックデザイナをインストールします。
次の手順からは、ロジックデザイナでの操作について説明します。

● 手順 2 プロジェクトの新規作成

制御アプリケーションは、「プロジェクト」単位で扱われます。
ロジックデザイナにより新規プロジェクトを作成し、プログラミングや各種属性を設定して、制御アプリケーションを開発します。
プロジェクトは、主に以下のものから構成されています。これらを作成／設定して、制御アプリケーションを作成します。

表　プロジェクトの主な構成要素

構成要素	概要
論理 POU	論理 POU は、各種開発言語でユーザが作成するプログラムです。
タスク	タスクは、論理 POU を実行形式（インスタンス化）にしたものです。
ライブラリ	ライブラリは、論理 POU を作成するときに使用する POU（プログラム・オーガニゼーション・ユニット）の集まりです。ライブラリには、標準、オプションパッケージ（アプリケーションポートフォリオ）、ユーザ作成ライブラリがあります。
変数	グローバル変数（デバイスラベル変数も含む）、ローカル変数があります。

● 手順 3 タスクの作成

タスクを作成します。タスクを作成するときに、実行周期を指定します。

補足

・ タスクを作成しないと、デバイスラベル変数の定義ができません。
・ デフォルトで定義されているタスクを用いる場合には、実行周期などタスクの定義が正しく設定されているか確認してください。

● 手順 4 デバイスラベル変数の定義

デバイスラベルとは I/O の各チャネルに割り振られたラベルです。このデバイスラベルに制御アプリケーションプログラムからアクセスするために、デバイスラベル変数を定義します。

参照

デバイスラベルとデバイスラベル変数については、「B4.3.4 入出力処理」「■デバイスラベルとデバイスラベル変数」参照してください。

● 手順 5 論理 POU の作成

論理 POU は、POU（プログラム・オーガニゼーション・ユニット）を組み合わせて作成します。POU はライブラリとして提供されています。
また、POU を組み合わせる開発言語としては、5 種類あり、用途に応じて使い分けられます。

● 手順 6 論理 POU のタスク割り付けとターゲット設定

作成した論理 POU をタスクに割り付けます（インスタンス化）。
さらに、ターゲット設定ダイアログで、IP アドレス（あるいはホスト名）を指定します。

● 手順 7 コンパイル、ダウンロード

作成したタスクをコンパイルし、FCN/FCJ にダウンロードできる形式にします。
作成した制御アプリケーションを FCN/FCJ にダウンロードします。

● 手順 8 デバッグ

制御アプリケーションが正常に動作するか、デバッグします。
主なデバッグ方法には、以下のものがあります。

- オンラインレイアウト表示とオンライン値表示
- ブレークポイントの設定
- ウォッチウインドウ
- ロジックアナライザ
- ソフトワイヤリング

● 手順 9 ドキュメンテーション

作成した制御アプリケーションをドキュメントとして、出力します。

B4.2.4 制御アプリケーション作成上の注意

● 共通の注意事項

- Target Setting ダイアログにて「タスクの実行時間がウォッチドッグタイムをオーバーした場合、アボートする」を“無効”としてください。
- タスクのウォッチドッグ時間には、タスク周期以上の値の設定を推奨します。
- POU のパッチは、デバッグ用としてのみ使用してください。

● CPU 二重化時の注意事項

- 制御アプリケーションを複数タスクで構成する場合、各タスクに割り付けたプログラムより同一のグローバル変数にアクセスしないでください。(*1)
- Java 機能は使用できません。
- CPU モジュールのシリアルポートは使用しないでください。
- 制御タスクには、少なくとも出力 1 点はデバイスラベル変数を割り付けてください。
- 制御ロジックの割り付けにはサイクリックタスクのみを使用し、システムタスクとデフォルトタスクは使用しないでください。
- SYSTEM 定義の“RAS Information”グローバル変数は、制御ロジックに使用しないでください。(*1)
- I/O 切り離し機能は使用しないでください。デバッグ時に I/O 切り離し機能を使用する場合は、シングル CPU 状態で使用してください。I/O 切り離し時に APC を実行した場合は、APC はエラー終了（I/O 切り離し中）し、システムアラームが出力されます。
- POU のパッチは使用しないでください。
- 制御側 CPU と待機側 CPU の同期処理が実行されるため、シングル CPU 構成時に比べ処理時間が長くなりますので、実機で動作パフォーマンスを確認してください。
- 制御側／待機側 CPU で値の違う変数（RAS などのシステム定義のグローバル変数）を用いてロジックを作成しないでください。これは、変数値が違うと制御側／待機側 CPU で違うロジックが動き、結果として待機側 CPU がダウンする場合があるためです。(*1)
  その例として、絶対時刻を参照する POU（PAS_GETTIME、PAS_GETTIME2）を制御演算や I/O 出力用のロジックに使用しないでください。２つの CPU の絶対時刻が異なる場合があり、絶対時刻を元に制御演算が実行された場合、演算結果に食い違いが生じる可能性があります。時刻による処理を実行させる場合は、タイマファンクションブロック（TOF、 TON、TP）を使用してください。

*1: CPU 二重化している FCN も、FCN/FCJ 間通信ファンクションブロックを用いるアプリケーションを作成すれば、FCN 内のグローバル変数にアクセスできます。

B4.3 制御アプリケーション

制御アプリケーションは、ロジックデザイナで作成されたソフトウェアです。制御アプリケーションを作成する際には、国際標準である IEC 61131-3 に準拠した 5 つの制御言語から用途や好みに応じて選択できます。
制御アプリケーションについて以下の項目を説明します。
・制御アプリケーションの内部構成
・制御アプリケーションのダウンロード

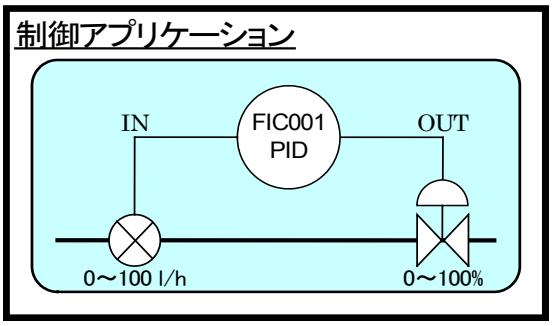


図　FCN/FCJ 制御アプリケーションのアプリケーションイメージ

補足

IEC 61131-3 に準拠した制御言語の概要および機能については、「新しいプログラマブルコントローラのプログラミング」（関口 隆 編：コロナ社）の参照を推奨します。

B4.3.1 制御アプリケーションの構成

制御アプリケーションは、以下のような構成になっています。

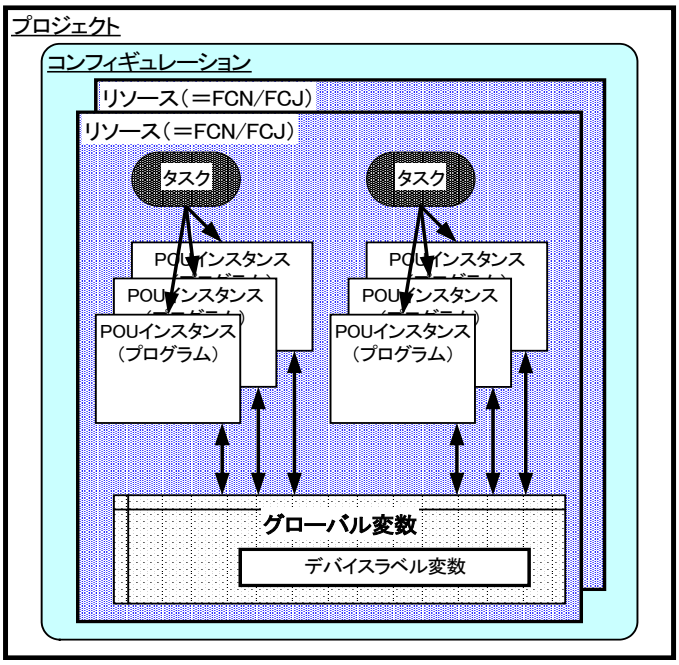


図　FCN/FCJ 制御アプリケーションの構成

■ プロジェクト（制御アプリケーション）

制御アプリケーションは、プロジェクトという単位で扱います。このプロジェクトは、開発ツールであるロジックデザイナを利用し、制御アプリケーションを開発する単位です。

■ コンフィギュレーション

プロジェクトには、１つ以上のコンフィギュレーション（定義）で構成されています。このコンフィギュレーションは、プロジェクト内のリソースを論理的に分類する抽象的な単位です。このため通常は１プロジェクトには、１つのコンフィギュレーションだけの定義を推奨します。（複数のコンフィギュレーションも使用できます。）

■ リソース

コンフィギュレーションは、1 つ以上のリソース（資源）から構成されています。リソースとは、1 台の FCN/FCJ に対応したプログラムです。複数のタスクおよび変数から構成されています。各々の FCN/FCJ に対応する実行形式の制御アプリケーションは、このリソース単位でダウンロードします。

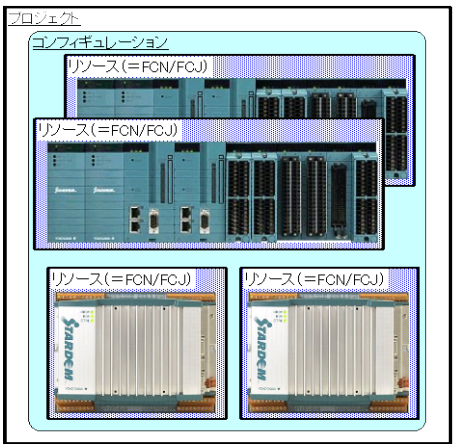


図　FCN/FCJ の制御アプリケーション構成（リソースのイメージ）

■ タスク

タスクは制御アプリケーションの実行スケジューリングの単位です。タスクの下には、1 つまたは複数のプログラムインスタンスが存在します。プログラム POU をインスタンス化（関連付け）すると、プログラムインスタンスとなります。タスクは、プログラムインスタンスを自身の動作設定に従い実行します。タスクには、以下のタイプがあります。

- CYCLIC
  一定周期で起動されるタスク（通常使用タイプ）。このタスクには、実行周期（msec）、ウォッチドッグ時間（msec）、プライオリティを指定します。
- DEFAULT
  他のタスクが停止しているときに常に動作しているタスク（特殊な場合以外使用しません）。このタスクにはウォッチドッグ時間（msec）を指定します。
- SYSTEM
  FCN/FCJ 起動時や、OS 内でエラーを認知したとき等に動作する特別なタスク（特殊な場合以外使用しません）。

補足

- ウォッチドッグ時間内にタスクが終了しない場合、エラーが発生します。
- プライオリティの等しいタスクは、30msec のタイムシェアリングで動作します。
- プライオリティの高いタスクは、プライオリティの低いタスクを中断してスケジューリングされます。
- タスクは、最大 16 個作成できます。

参照

タスクの詳細な動作については、「B4.3.3 タスクのスケジュール」を参照してください。

■ 論理 POU

論理 POU は、制御アプリケーションを構成する機能部品で、以下の 3 種類のタイプがあります。

- プログラム
- ファンクションブロック
- ファンクション

論理 POU は、通常ファンクションあるいはファンクションブロックの組み合わせで構成されます。ファンクションブロックの中でも別のファンクションブロックを使用できるので、階層的な制御アプリケーションを構築できます。

論理 POU のロジックは、IEC 61131-3 が規定する 5 種類の開発言語を用いて記述できます。ユーザは作成するアプリケーションに最適な言語を適宜選択して、論理 POU を作成できます。

- SFC (Sequential Function Chart):シーケンシャル・ファンクション・チャート
- FBD (Function Block Diagram):機能ブロック図
- IL (Instruction List):命令リスト
- ST (Structured Text):構造化テキスト
- LD (Ladder Diagram):ラダー図

● プログラム POU

プログラム POU は、論理 POU の階層構成において最上位に位置する論理 POU です。このプログラム POU の制御タスクへのインスタンス化（割り付け）により、記述されている処理が動作します。

プログラム POU を作成する場合は、ファンクション POU またはファンクションブロック POU を利用できます。

● ファンクション POU

ファンクション POU は、複数の入力と、ただ 1 つの出力を持つ POU です。他 POU で定義すれば使用できます。本 POU は内部メモリを持っていません。このため、入力値が同じ場合は常に同じ出力値が得られます。

ファンクション POU を作成する場合は、他のファンクション POU を利用できます。

補足

ファンクション POU 内ではグローバル変数は使用できません。

● ファンクションブロック POU

ファンクションブロック POU は、複数の入力と出力を持てる POU です。他のファンクションブロック POU またはプログラム POU で定義すれば使用できます。本 POU は、内部メモリを持ちます。本 POU を他 POU で定義するとにより、インスタンス化（実体化）され動作できるようになります。
ファンクションブロック POU を作成する場合は、他のファンクションブロック POU およびファンクション POU を利用できます。

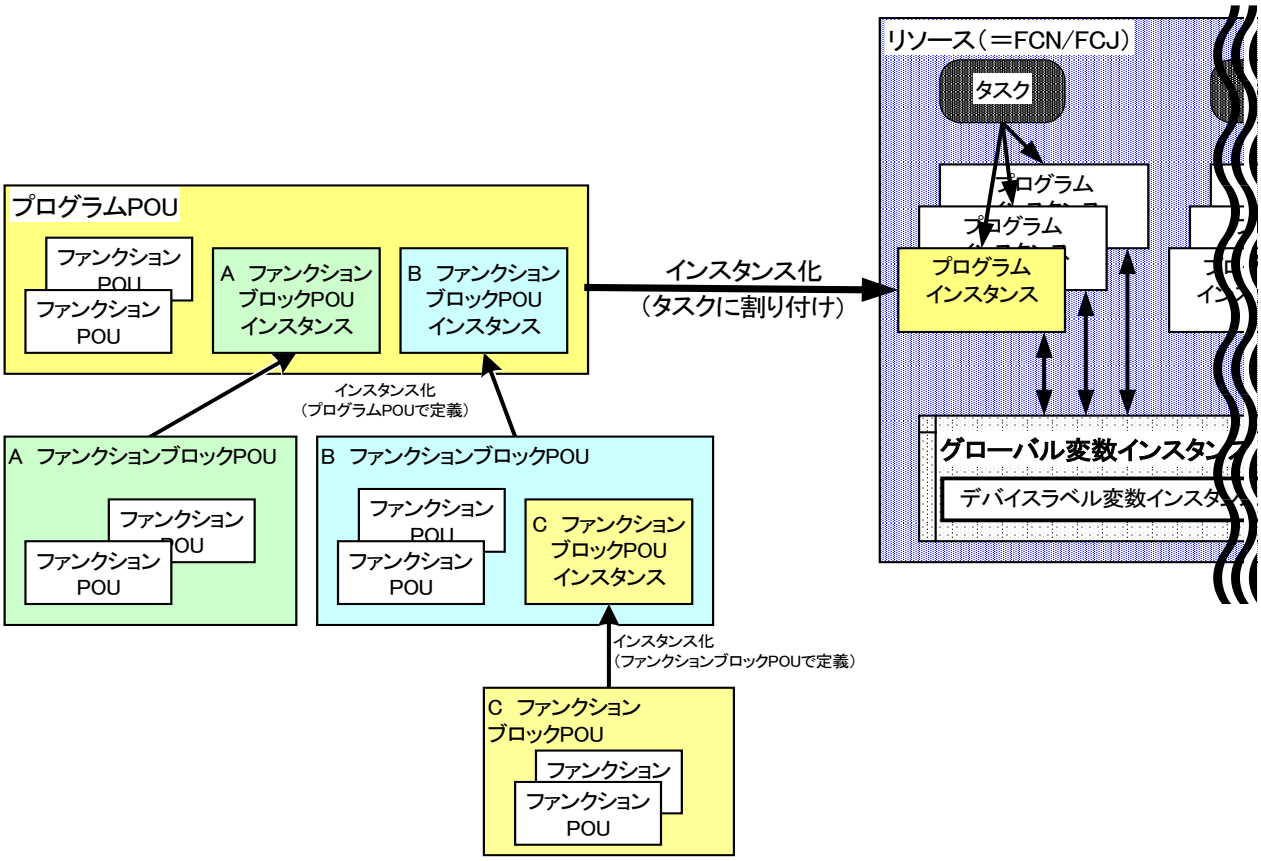


図　FCN/FCJ 制御アプリケーションの構成（イメージ）

■ ローカル変数とグローバル変数

制御アプリケーションで使用される変数は、そのスコープ（有効範囲）の観点から、以下のように分類できます。

● ローカル変数

スコープが論理 POU である変数です。
ローカル変数を定義した論理 POU 内でのみ、使用可能です。

● グローバル変数

スコープがリソースである変数です。リソース内で動作するすべての POU インスタンスからアクセス可能です。

- デバイスラベル変数
  デバイスラベル変数は I/O を表現する特別なグローバル変数です。
  制御アプリケーション内部での扱いは、通常のグローバル変数と同等です。

補足

制御アプリケーションで使用される変数は、スコープ以外にもいくつかの属性を持ちます。
以下に主だったものを示します。

- 「保持」属性
  この属性をオンに設定した変数に関しては、FCN/FCJ の電源断時やウォームスタート時に変数値が保持されます。
- 「OPC」属性
  この属性をオンに設定した変数に関しては、FCN/FCJ 外部や FCN/FCJ 内部の Java アプリケーションから、変数値の参照・設定が可能になります。

B4.3.2 制御アプリケーションのダウンロード

制御アプリケーションは、汎用 PC で動作するロジックデザイナで作成します。作成した制御アプリケーションは、FCN/FCJ にダウンロードします。ダウンロードする単位は以下の 2 種類です。

- **リソース単位**
- **ソースプロジェクト単位**

■ リソース単位のダウンロード

制御アプリケーション中のリソースは、各々の FCN/FCJ に対応しています。リソースをダウンロードすると、実行可能形式としてプログラムがダウンロードされます。

この制御アプリケーションのリソースをダウンロードする先は、以下の 2 通りの場所があります。

1. 揮発性メモリ
   FCN/FCJ の揮発性メモリにリソースをダウンロードします。制御アプリケーションのデバッグ時にのみ使用します。
   FCN/FCJ が再起動されると揮発性メモリへダウンロードした情報は、すべてクリアされます。
2. FCN/FCJ システムカード
   デバッグが完了した制御アプリケーションは、FCN/FCJ システムカード上にダウンロードしてください。FCN/FCJ を再起動すると FCN/FCJ システムカード上の制御アプリケーションが揮発性メモリ上にコピーされ、動作を開始します。

補足

ロジックデザイナで制御アプリケーションを FCN/FCJ システムカードにダウンロードする場合は、「ブートプロジェクト」を選択し、ダウンロードしてください。

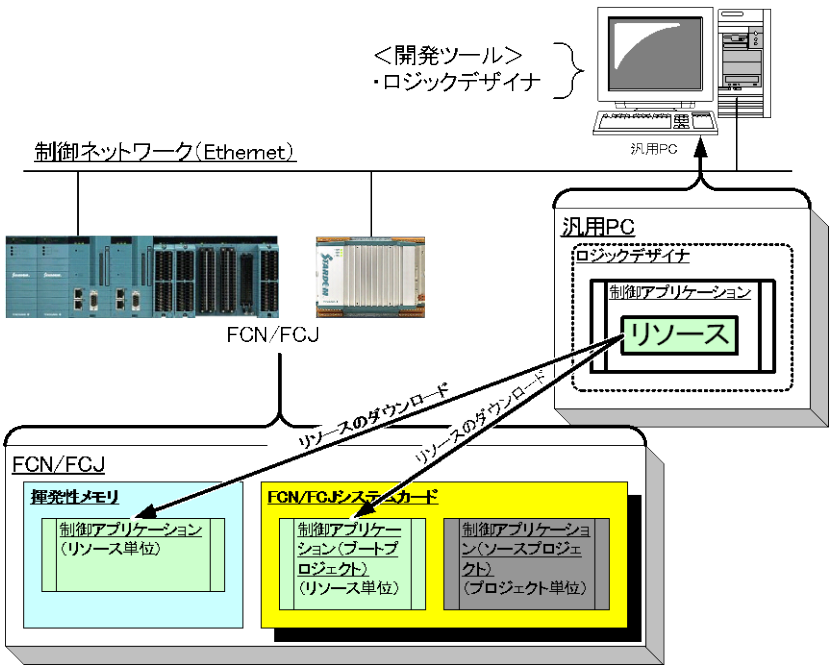


図 制御アプリケーションのダウンロード先

■ ソースプロジェクトのダウンロード

制御アプリケーション（プロジェクト）のすべてを FCN/FCJ システムカード上にダウンロードします。これは、制御アプリケーションのバックアップを目的としています。このファイルのアップロードも可能です。

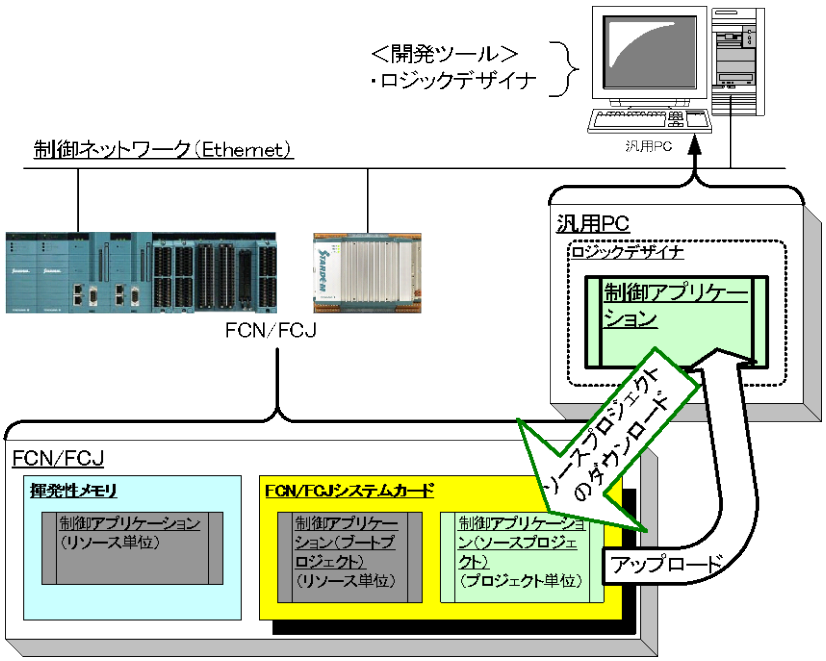


図　制御アプリケーションのダウンロード先

参照

ロジックデザイナには、FCN/FCJ の制御動作を継続しながら制御アプリケーションを変更できるオンラインダウンロード機能があります。オンラインダウンロードについては、「B6.3 オンラインダウンロード」を参照してください。

B4.3.3 タスクのスケジュール

制御アプリケーションで作成される CYCLIC タイプのタスク（プログラム）は、FCN/FCJ 上で同時に実行できます。このときのタスクは、下記のルールで動作します。

- **プライオリティの高いタスクが優先して動作します（プライオリティの低いタスクは中断されます）。**
- **プライオリティの等しいタスクは、30msec のタイムシェアリングで動作します。**
- **各タスクが実行を開始したときに、そのタスクに割り振られた I/O の入力処理を実行します。**
- **各タスクが実行を終了するときに、そのタスクに割り振られた I/O の出力処理を実行します。**
- **タスクが指定時間内に実行できない場合には、次の周期はスキップされます。**
- **タスクの実行周期をウォッチドッグ時間で監視できます。**

これらを、いくつか例を示しながら解説します。

■ プライオリティによるタスクの動作

プライオリティの高いタスクが、プライオリティの低いタスクを中断して動作します（最高プライオリティ：0、最低プライオリティ：31）。
3 つのタスクが下記条件で動作したときの動作例を示します（各タスクの実行時間は常に一定と仮定します）。

表　タスクの実行例

タスク名	プライオリティ	実行周期	実行時間
Task-A	高（10）	100 ms	50 ms
Task-B	中（15）	300 ms	100 ms
Task-C	低（20）	1000 ms	100 ms

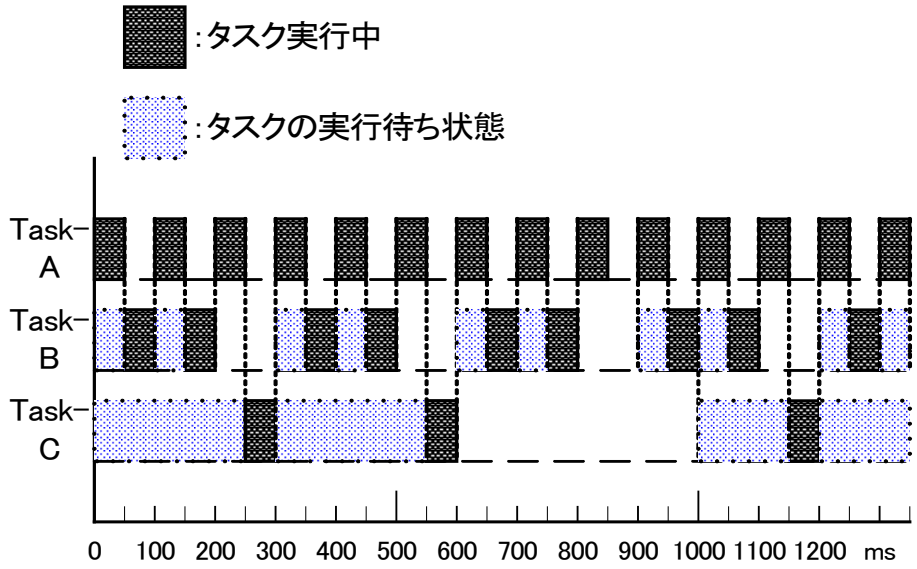


図　プライオリティによるタスクの実行順序

上図のように、プライオリティの高い Task-A が動作すると、他のタスクは常に実行待ちとなります。

■ プライオリティの等しいタスクの実行権

プライオリティの等しいタスク間では、30msec ごとに実行権が切り換わります。
2 つのタスクが下記条件で動作したときの動作例を示します。
（各タスクの実行時間は常に一定と仮定します。）

表　タスクの実行例

タスク名	プライオリティ	実行周期	実行時間
Task-A	中（15）	300msec	120msec
Task-B	中（15）	300msec	90msec

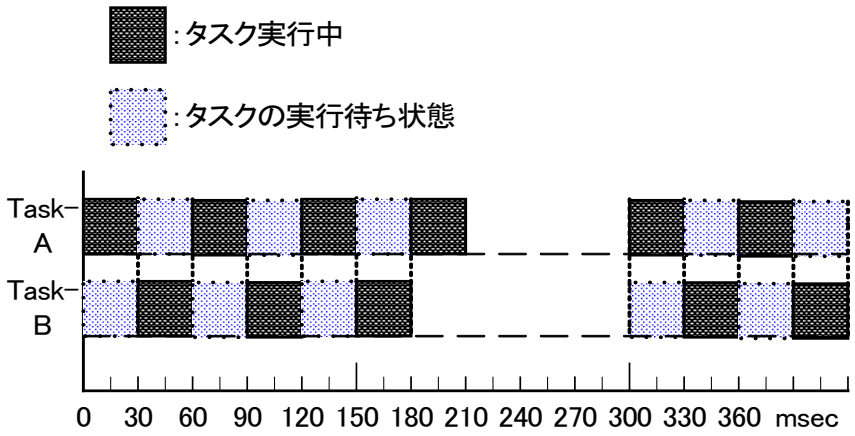


図　プライオリティの等しいタスク実行

上図のように、プライオリティの等しいタスクが重なってスケジュールされると、それぞれのタスクは 30msec ずつ動作し、その後で実行待ち状態に入ります。

■ タスクの I/O 処理タイミング

1 つのタスクが 1 周期を実行するには、前処理として I/O の入力処理、後処理として I/O の出力処理を実行します。タスクに割り付けられた I/O を対象とします。

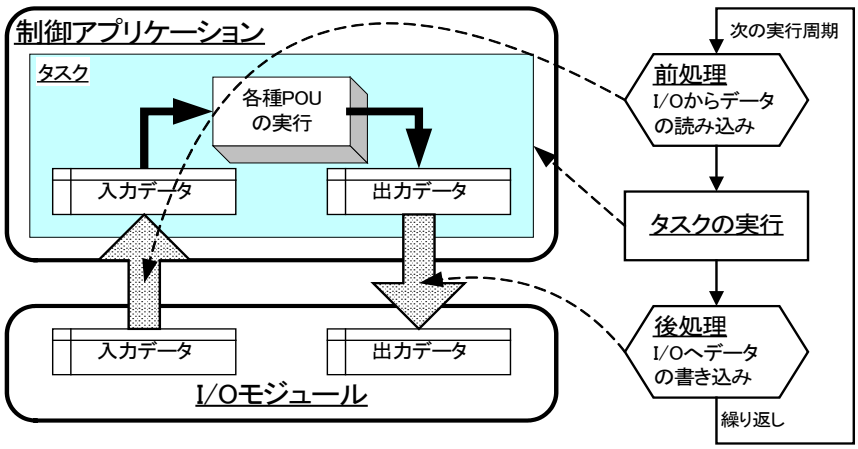


図　タスクの前処理と後処理

補足

ロジックデザイナでデバイスラベル変数として I/O を定義します。この I/O を指定するときに、I/O を使用するタスクを指定します。このタスクを実行するときに対応する I/O 処理を実行します。

■ タスクが実行周期以上に実行時間を費やしたときの動作

タスクの実行時間は、常に一定とは限りません。プログラムしだいでは実行時間が長くなる場合があります。
この場合、タスクの実行周期より実行時間が長くなると、次の実行周期はスキップされます。

表　タスクの実行例

タスク名	実行周期	実行時間
Task-A	0.5sec	0.2～0.6sec まで変化

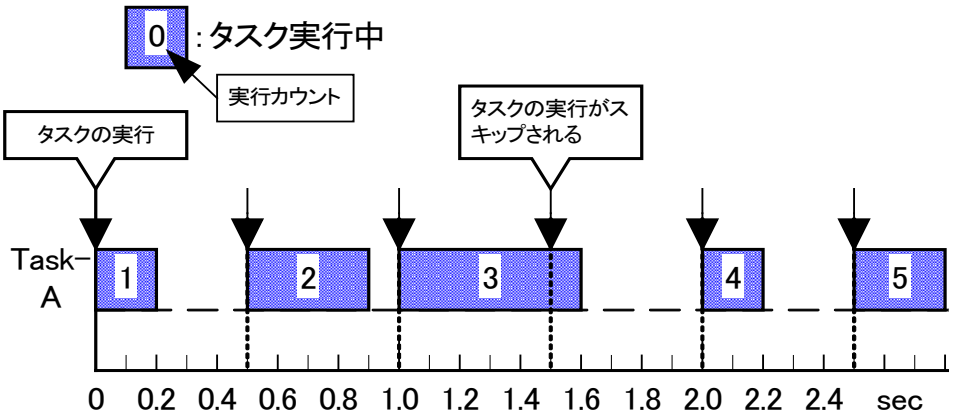


図　タスクが実行周期以上に実行時間を費やしたときの動作

上図のように、タスクが実行周期内に処理を終了できないと、次の実行周期では、タスクが実行されません。

補足

他の高いプライオリティのタスクが動作していて、タスクが動作できないときにも同様に動作します。

■ タスクの実行周期とウォッチドッグ時間

タスクが一定時間以内（ウォッチドッグ時間）に、実行されたか監視できます。
タスクの実行周期とウォッチドッグ時間の関係例を以下に示します。

● タスクの実行時間がウォッチドッグ時間を越えた場合

表　タスクの実行例

タスク名	タスクの実行周期	タスクの実行時間	タスクのウォッチドッグ時間
Task-A	0.5sec	0.2～0.6sec まで変化	0.5sec (*1)

*1:　タスクの実行周期とウォッチドッグ時間が同一の場合

1. ターゲット設定ダイアログで「タスクの実行時間がウォッチドッグタイムをオーバした場合、アボートする」がチェックされている場合

タスクがウォッチドッグ時間以上に実行時間を費やし、ターゲット設定ダイアログで「タスクの実行時間がウォッチドッグタイムをオーバした場合、アボートする」がチェックされていると、エラーが発生し実行タスクは停止します。

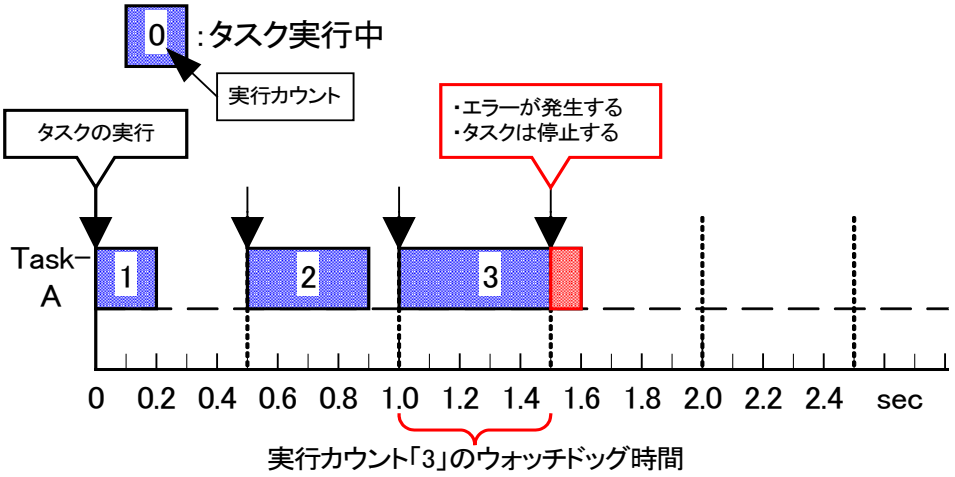


図　ウォッチドッグ時間以上にタスクが実行時間を費やし、タスクが停止する動作例

2. ターゲット設定ダイアログで「タスクの実行時間がウォッチドッグタイムをオーバした場合、アボートする」がチェックされていない場合

タスクがウォッチドッグ時間以上に実行時間を費やし、ターゲット設定ダイアログで「タスクの実行時間がウォッチドッグタイムをオーバした場合、アボートする」がチェックされていない場合、エラーが発生し次の実行周期ではタスクが実行されませんが、タスクは停止しません。

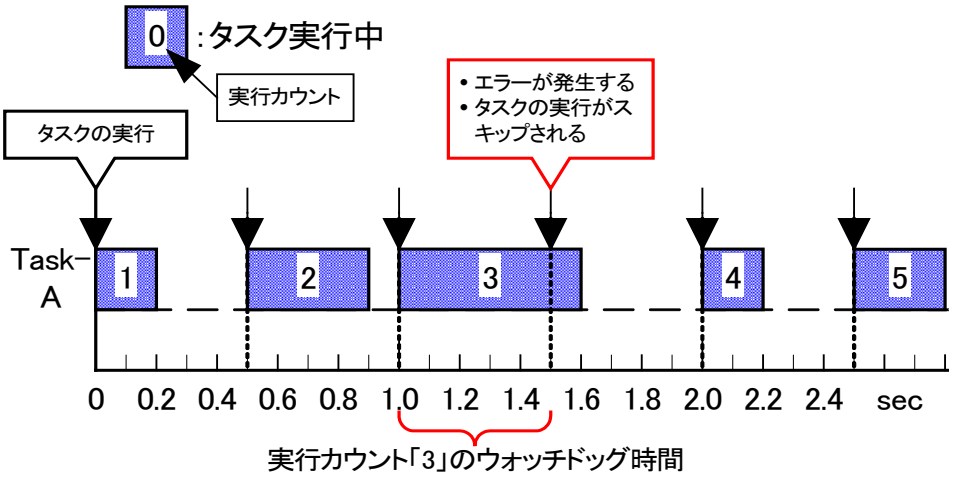


図　ウォッチドッグ時間以上にタスクが実行時間を費やすが、タスクは停止しない動作例

● タスクの実行時間がウォッチドッグ時間を越えない場合

タスクが実行周期以上に実行時間を費やしたが、ウォッチドッグ時間以内にタスクが実行されたとき、次の実行周期ではタスクが実行されません。しかし、ウォッチドッグ時間を越えていないのでエラーは発生しません。

表　タスクの実行例

タスク名	タスクの実行周期	タスクの実行時間	タスクのウォッチドッグ時間
Task-A	0.5sec	0.2～0.6sec まで変化	1.0sec (*1)

*1:　タスクの実行周期の 2 倍がウォッチドッグ時間の場合

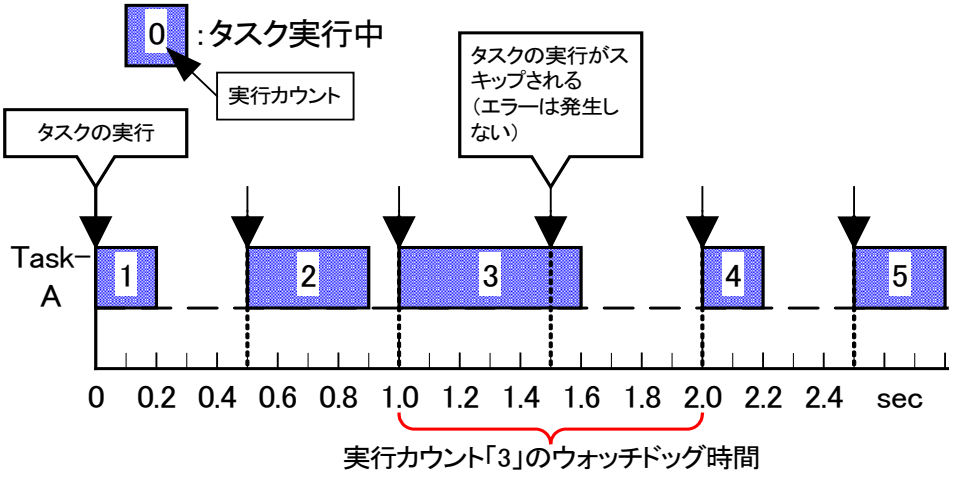


図　ウォッチドッグ時間以内にタスク実行された動作例

■ CPU オーバーロード

タスクの実行時間と実行周期が一致するなど CPU オーバーロード状態になると、空き時間に実行する通信処理や Java アプリケーションが動作できなくなります。
この CPU オーバーロード状態が 10 秒間継続すると CPU オーバーロードエラーが発生します。
CPU オーバーロードエラーが発生する場合は、制御アプリケーションなどを見直してください。

補足

CPU オーバーロードエラーが発生すると、下記のログが FCN/FCJ のログファイルに保存されます。
例）CPU オーバーロードのログ

```
2007/06/15 09:20:25、……、Msg=Exception(CPU Overload) in Task0、
```

● CPU オーバーロードエラーが発生する状況

CPU オーバーロードエラーが発生する主な状況には、下記があります。

- パターン 1：タスクの実行時間と実行周期が一致する場合
  タスクの実行時間と実行周期が一致すると CPU の空き時間が 0 となります。
  タスクの実行周期とウォッチドッグ時間が一致するため、ウォッチドッグエラーも発生しません。
  この状態が 10 秒以上継続すると CPU オーバーロードエラーが発生します。

表　パターン 1：タスクの実行時間と実行周期が一致する場合の実行例

タスク名	タスクの実行周期	タスクの実行時間	タスクのウォッチドッグ時間
Task-A	1.0 sec	1.0 sec	1.0sec

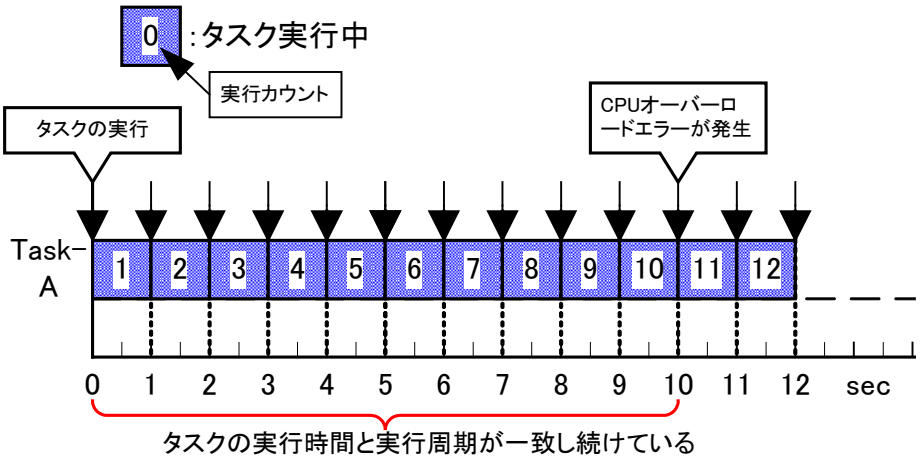
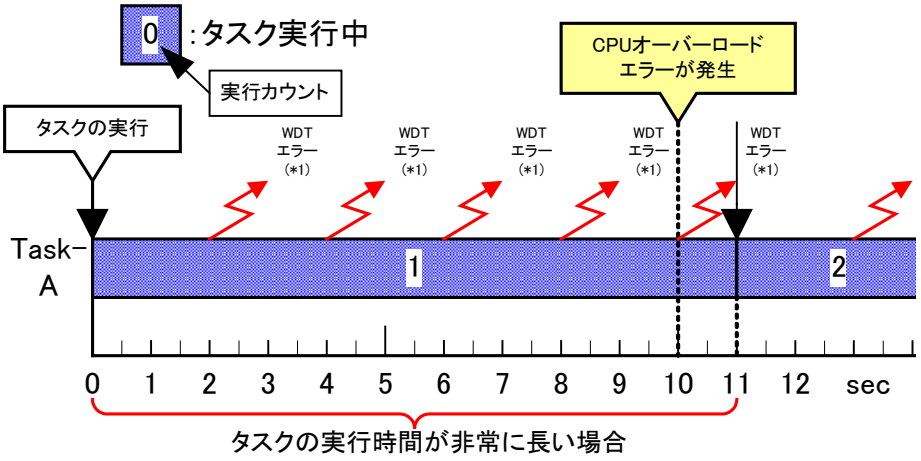


図　パターン 1：タスクの実行時間と実行周期が一致する場合の動作例

・ パターン 2：タスクの実行時間が長い場合
1 回のタスクの実行時間が 10 秒以上継続すると、CPU オーバーロードエラーが発生します。
ウォッチドッグ時間ごとにウォッチドッグエラーが発生します。

表　パターン 2：タスクの実行時間が長い場合の実行例

タスク名	タスクの実行周期	タスクの実行時間	タスクのウォッチドッグ時間
Task-A	2.0 sec	11.0 sec	2.0sec



*1:　WDT エラー：ウォッチドッグエラーが発生

図　パターン 2：タスクの実行時間が長い場合の動作例

補足

CPU オーバーロードエラーが発生する要因として下記のものがあります。
・ 制御アプリケーションの繰り返し処理が終わらなくなった。
（例：ST 言語の FOR 文の処理や、ジャンプとラベルの処理の間違いにより、アプリケーションが無限ループの状態になった、など）
・ 制御アプリケーションより優先順位の高いタスクの処理が継続して実行された。
（例：Ethernet 上の機器がブロードキャスト通信を続けている、など）

● CPU オーバーロード時の動作

CPU オーバーロード状態では、下記の機能が動作しません。

1. FCN/FCJ の Ethernet 機能
・ VDS/ASTMAC のデータサーバとの通信
・ FCN/FCJ 間通信
・ Modbus 通信など Ethernet 通信またはシリアル通信を用いた各機器間の通信
・ ロジックデザイナやリソースコンフィギュレータからの操作／設定
など
2. FCN/FCJ の Java 機能
・ InfoWell 機能
・ ユーザ作成の Java アプリケーション
など

B4.3.4 入出力処理

ハードウェア（入出力端子）とソフトウェア（制御アプリケーション）とを機能的に接続する入出力処理の概念についてと、入出力データの更新タイミングについて説明します。

■ デバイスラベルとデバイスラベル変数

FCN/FCJ が入出力処理を実行するとき、デバイスラベルとデバイスラベル変数を指定します。この概念について説明します。

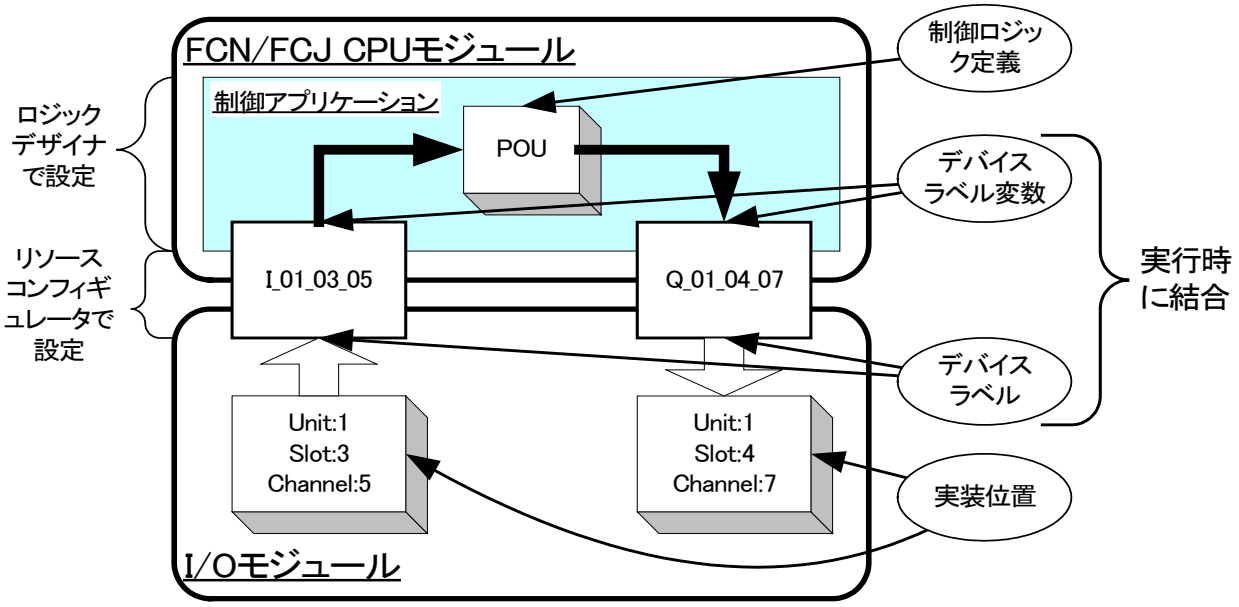


図　デバイスラベルとデバイスラベル変数の関係

● デバイスラベル

デバイスラベルは、各物理 I/O に名前を付けたものです。デバイスラベルは、リソースコンフィギュレータで設定および変更できます。

補足

- デバイスラベルは、英数字、“_”（アンダーバー）からなる英字または“_”で始まる最大 16 文字の文字列です。
- デバイスラベルは、初期値として名前が割り付けられています。変更せずにそのままでも使用できます。初期値は以下のようになっています。
  FCN：「x_uu_ss_cc」（x:信号種別、uu:ユニット番号、ss:スロット番号、cc:チャネル番号）
  FCJ：「x_d_cc」（x:信号種別、d:デバイスタイプ、cc:チャネル番号）
  x：信号種別（I：入力信号、Q：出力信号）
  d：デバイスタイプ（A：アナログ入出力、D：デジタル入出力）

● デバイスラベル変数

デバイスラベル変数は、ロジックデザイナで制御アプリケーションを作成するときに指定する論理的な I/O の名前です。論理的な名前であるため、物理 I/O を意識せずに制御アプリケーションを作成できます。
デバイスラベル変数は、ロジックデザイナで指定します。

● デバイスラベルとデバイスラベル変数の結合

デバイスラベルとデバイスラベル変数は、同一の名前のものが結合されます。結合の時期は、制御アプリケーションの実行開始時です。
このように、制御アプリケーションと物理 I/O は、実行時に結合されるため、制御アプリケーションを I/O 定義と独立して開発できます。

■ 入出力データの更新タイミング

CPU モジュール内では、

1. I/O モジュールからのデータ入力
2. 制御アプリケーションの実行
3. I/O モジュールへのデータ出力

の順に同期して処理が実行されていますが、I/O モジュールでは、FCN/FCJ の CPU モジュールとは非同期に入出力処理が実行されます。その結果、タイミングによっては最大 1 周期分のずれが生じる場合があります。
このため、I/O モジュールのデータ収集周期を考慮した制御アプリケーションを作成する必要があります。

参照

制御アプリケーションの入出力応答時間については、「B4.5.4 入出力処理のパフォーマンス」を参照してください。

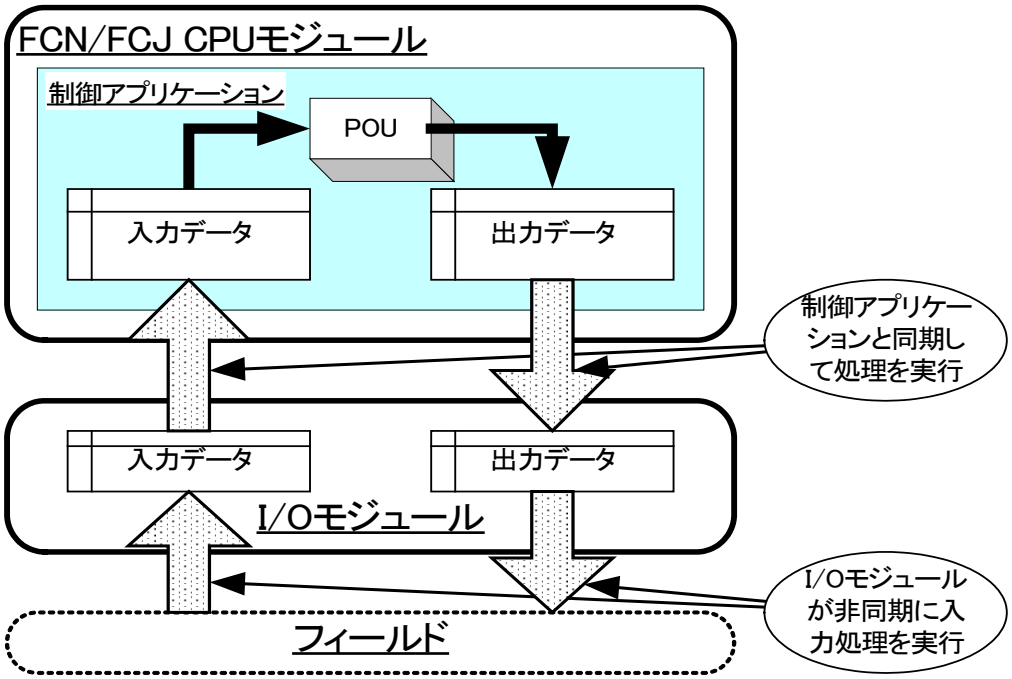


図　入出力データの更新タイミング

B4.3.5 保持変数（リテインデータ）

保持変数（リテインデータ）は、変数の「保持」属性がオンに指定されている変数です。通常、FCN/FCJ の電源断時にも値を保持したい変数を、保持変数に指定します。ここでは、保持変数に関して、下記の項目について解説します。

- **保持変数が確保される場所**
- **保持変数値の継続性**
- **変数の初期化とスタートモードの関係**

補足

PAS POU のアクセスパラメータとエンジパラメータは、デフォルトで変数の「保持」属性がオンに指定されています。

■ 保持変数が確保される場所

保持変数は、下記の場所に確保されます。

- 不揮発メモリ上（CPU モジュールの設定による）
- 揮発メモリ上（CPU モジュールの設定による）
- FCN/FCJ システムカード（保持変数のバックアップ）

● 不揮発性メモリ上

CPU モジュールの設定で「リテインデータのハードバックアップを有効にする」が選択されている場合は、現在動作している制御アプリケーションのリテインデータを不揮発性メモリ上に保持しています。

補足

不揮発性メモリは、揮発性メモリに比べメモリのアクセス速度が遅いため、リテインデータを使用する POU があると、制御アプリケーションの実行速度が遅くなります。

● 揮発性メモリ上

CPU モジュールの設定で「リテインデータのハードバックアップを有効にする」が選択されていない場合は、現在動作している制御アプリケーションのリテインデータを揮発性メモリ上に保持しています。揮発性メモリの保持変数は、電源を切るとシステムデフォルト値が設定されます。
ただし、制御アプリケーションで初期値が設定されている保持変数は、その初期値が保持変数に設定されます。

補足

- FCN/FCJ システムカード上にバックアップが存在する場合は、次回制御を開始する際（電源再投入時、またはウオームスタート時）その内容が復元されます。
  従って、CPU モジュールの設定で、「リテインデータのハードバックアップを有効にする」が選択されていない場合は、保持変数値の継続性を確保するため、定期的に保持変数をバックアップしてください。
- 「リテインデータのハードバックアップが有効にする」が選択されている場合でも、次回制御を開始する際（電源再投入時、またはウオームスタート時）、アプリケーションの要求するリテインの構造が不揮発性メモリ上のリテインデータと一致しない場合は、FCN/FCJ システムカード上のリテインデータが復元されます。

● FCN/FCJ システムカード

メモリ上のリテインデータを FCN/FCJ システムカードにバックアップできます。以下の方法でバックアップできます。

- インタフェースフラグ（リテインデータのセーブスイッチ：GS_RETAIN_SV_SW）をオンにして実行します。(*1) (*2)
- FCN/FCJ メンテナンス画面の「Save Retain Data」よりバックアップします。

*1: リテインデータのセーブスイッチをオンにするには以下の方法があります。
・ロジックデザイナのデバッグモードを用いて、手動でオンにします。
・インタフェースフラグをオンにするアプリケーションを作成し、実行します。

*2: リテインデータのセーブスイッチがオンになると、リテインデータのセーブを開始します。
リテインデータのセーブが終了すると自動的にオフになります。

■ 保持変数値の継続性

保持変数のバックアップを復元する処理は、保持変数領域（不揮発性／揮発性メモリ：CPU モジュールの設定に従う）の状態や、ロードされている制御アプリケーションの状態に深く関係があります。ここではバックアップの復元処理を含め、保持変数値の継続性について説明します。

■ 保持変数領域が初期状態のとき

保持変数領域が初期状態のとき（CPU モジュールが初期状態の場合、不揮発性メモリ保持用バッテリを脱着した場合、CPU モジュールの「リテインデータのハードバックアップを有効にする」設定がオフで、FCN/FCJ の電源が切られた場合など）に、FCN/FCJ の電源が投入されると、保持変数の扱いは以下のようになります。

● FCN/FCJ システムカードに保持変数のバックアップが無い場合

制御アプリケーションで初期値が設定されている保持変数は、その初期値が保持変数に設定されます。それ以外は、システムのデフォルト値が設定されます。

● FCN/FCJ システムカードに保持変数のバックアップが有る場合

以下の手順でバックアップの内容が復元されます。

1. 各保持変数は、システムのデフォルト値が設定されます。
   制御アプリケーションで初期値が設定されている保持変数は、その初期値が保持変数に設定されます。
2. バックアップの中に保持変数値が保存されている場合は、その値が復元されます。

すなわち、バックアップの内容を最大限復元します。

■ 保持変数領域が初期状態でないとき

保持変数領域が初期状態でないとき（CPU モジュールの「リテインデータのハードバックアップを有効にする」設定がオンで、FCN/FCJ の電源が切られた場合など）に、FCN/FCJ の電源が投入されるか、またはウォームスタートされると、保持変数の扱いは以下のようになります。

● 制御停止後、保持変数の構成（数、データ型など）に変更が無い状態での制御再開時

制御停止後、保持変数の数やデータ型に変更が無い状態（制御アプリケーションを一切変更していない場合、あるいは保持変数の数やデータ型には関わらない変更のみの場合）で制御を再開した場合は、保持変数領域の内容をそのまま利用し、制御アプリケーションの動作を開始します。

● 制御停止後、保持変数の構成（数、データ型など）を変更した状態での制御再開時

制御停止後、保持変数の数やデータ型を変更した状態で制御を再開した場合、保持変数の内容は、初期化されます。その後、以下の手順でバックアップの内容が復元されます。

1. 各保持変数は、システムのデフォルト値が設定されます。
   制御アプリケーションで初期値が設定されている保持変数は、その初期値が保持変数に設定されます。
2. バックアップの中に保持変数値が保存されている場合は、その値が復元されます。
   バックアップが無い場合には、復元処理が実行されません。

■ 変数の初期化とスタートモードの関係

ロジックデザイナを用いて、デバッグ時に指定する制御エンジンの 3 つのスタートモード（コールド/ウォーム/ホットスタート）があります。この機能で最も異なるのは、変数の初期化に関する処理です。以下にスタートモードと変数の初期化について説明します。

表　制御エンジンのスタートモード

スタートモード	非保持変数	保持変数	備考
コールドスタート	初期化します。 初期値を指定されたものは初期値を設定。 初期値が指定されていないものはデフォルト値（0）を設定。	初期化します。 初期値を指定されたものは初期値を設定。 初期値が指定されていないものはデフォルト値（0）を設定。	
ウォームスタート	初期化します。 初期値を指定されたものは初期値を設定。 初期値が指定されていないものはデフォルト値（0）を設定。	値を保持。(*1)	(*2)
ホットスタート	値を保持。	値を保持。	制御アプリケーションの実行停止後、リセット処理や制御アプリケーションのダウンロード処理を実行すると、ホットスタートはできなくなります。

*1:　不揮発性メモリ上に配置される保持変数の構成に変更があった場合は初期化されます。
　　リテインデータがセーブされている場合は、初期化の後セーブされているデータがリストアされます。
*2:　揮発性メモリ上に保持変数が確保されていてウォームスタートした場合、値を保持します。

補足

- 電源オン時（ブートプロジェクトを使った起動時）の場合は、常にウォームスタートとなります。
- オンラインダウンロード機能を用いて制御アプリケーションを変更しても、非保持変数、保持変数は、値を保持します。
  ただし、変数名やデータ型が変更された変数は、値を保持しません。

B4.4 機能仕様

FCN/FCJ の CPU 機能仕様は、下表のとおりです。また、FCN/FCJ 1 台あたりの制御アプリケーション容量の目安、および最大構成の一例も下表に記載しています。

表　CPU 機能仕様

実行速度	約 50 μ sec/Kstep（IL 言語使用時）
制御アプリケーション数	最大 16 タスク
タスクプライオリティ	指定可（最大 16 段階）
タスク実行周期	10msec 以上（10msec 単位）

表　機能仕様

制御アプリケーションプログラム容量	最大 3MB（約 400Kstep：IL 言語使用時）
データ領域 (*1)	最大 8MB
リテインデータ領域 (*2) (*3)	最大 410KB
Java アプリケーション容量	最大 32MB

*1:　電源 OFF 時にデータは保持されません。
*2:　電源 OFF 時にデータは保持されます。（制御アプリケーションのチューニングパラメータ格納用としても使用されます）
*3:　総容量の内、350 KByte が制御アプリケーションで使用可能です。

参照

制御アプリケーションの容量については「B4.5.2 制御アプリケーション容量の算出」を参照してください。

■ 最大容量の目安

● 制御計器数　：512 個

- 連続制御（指示計、調節計など）：128 個
- その他（演算、スイッチ計器、PLC 通信など）：384 個

● シーケンス　：ラダー換算：180Kstep

- シーケンステーブル換算：128 枚（Condition:32、Action:32）

■ 最大構成例：

- AI：96 点、AO：32 点、DI：256 点、DO：256 点
- PID ループ：32 ループ
- シーケンステーブル換算：128 枚
- 制御周期：1 秒

■ システムカード仕様

表　システムカード仕様

FCN/FCJ システムカード	128 MByte タイプ	512 MByte タイプ(*1)
システム領域	32 MByte	
ユーザアプリケーション領域(*2)	96 MByte	480 MByte

*1:　データを FCN/FCJ システムカードに保存する Java アプリケーションを実現する場合は、512MByte のシステムカードを推奨します。
*2:　ユーザアプリケーション領域には、制御アプリケーションと Java アプリケーションとデータの各ファイルが含まれます。制御アプリケーションのファイルサイズは、最大 10 MByte です。

B4.5 パフォーマンス

本章では、FCN/FCJ の制御アプリケーションの実行速度などを解説します。

B4.5.1 FCN/FCJ のスキャン周期と制御周期

FCN/FCJ の制御機能は、タスクごとに定義された一定の周期で実行されます。この周期をスキャン周期と呼びます。I/O モジュールから CPU への読み込み処理、CPU から I/O モジュールへの書き込み処理は、このスキャン周期ごとに実行されます。実際の制御演算を実行する NPAS POU 処理では、スキャン周期の整数倍のタイミングで「制御演算」を実行します。この制御演算を実行する周期を制御周期と呼びます。制御周期は「NPAS_POU ごと」あるいは、「FCN/FCJ の一括設定」が可能です。なお、入力信号の異常などへ迅速に対応するため、また HMI へ表示するため、警報処理および入出力の処理は、スキャン周期ごとに実行されます。

表　制御機能の動作（CPU）

No.	処理			実行タイミング	備考
①	I/O モジュールから CPU への読み込み処理			スキャン周期ごと	ドライバプログラム
②	NPAS_POU 処理	(1)	入力処理	スキャン周期ごと	
		(2)	制御演算処理	制御周期ごと	
		(3)	警報処理	スキャン周期ごと	
		(4)	出力処理	スキャン周期ごと	
③	CPU から I/O モジュールへの書き込み処理			スキャン周期ごと	ドライバプログラム

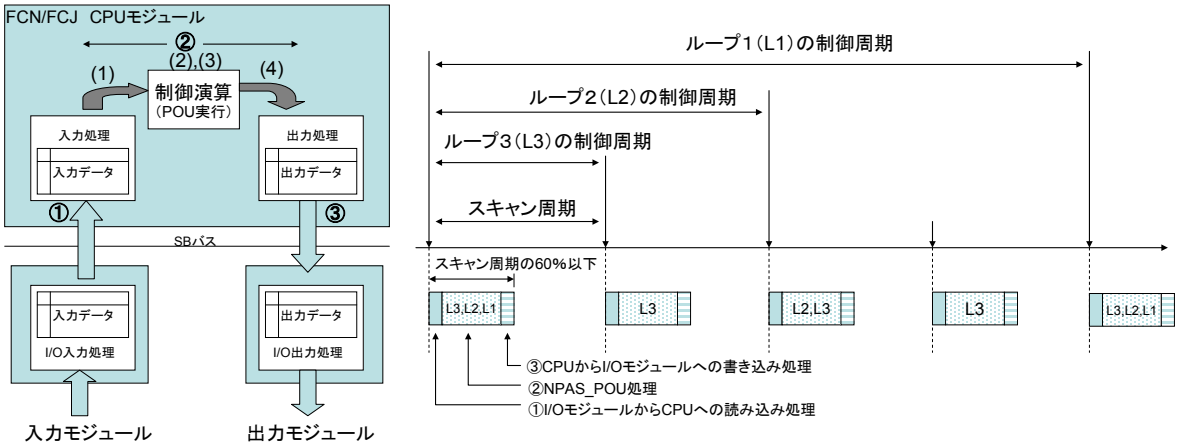


図　CPU の制御機能の動作とスキャン周期、制御周期

● スキャン周期の設定

スキャン周期は、タスクを作成するときに決定します。スキャン周期は 10ms 以上 10ms 単位で設定できます。プロセス特性やパフォーマンスなどを考慮して、最適なスキャン周期を決定します。

B4.5.2 制御アプリケーション容量の算出

制御アプリケーションプログラムの最大容量は、以下の 3 種類の容量によって規定されます。

1. **プログラム容量：3 MByte**
   NPAS_POU などで記載された制御プログラム本体を格納するメモリ部のサイズ。（IL 言語換算で約 400 Ksteps）
2. **ADLST 容量：4MByte**
   ADLST とはアプリケーションデータリストの略。ADLST 容量とは変数領域のサイズ。
3. **リテインデータ容量：350KByte**
   リテインデータとは保持変数の略で停電時に保持されるデータ。410KByte の総容量のうち、350KByte が制御アプリケーションで使用可能。

■ NPAS_POU を用いる場合の概算方法

上記 3 種類の容量制約のうち、ADLST 容量およびリテインデータ容量の制約の影響が大きくなります。（NPAS_POU を ADLST 容量の上限まで使用してもプログラム容量の 15%程度のためです。）このため下記のように ADLST 容量とリテインデータ容量を算出して確認します。

● ADLST 容量の算出

ADLST 容量は下記の計算式を用いて、下表を参考にして算出します。

ADLST 容量＝固定サイズ（約 17KByte）＋NPAS_POU の ADLST サイズ合計
＋OPC のチェックが有る変数の ADLST サイズ合計

表　NPAS POU の ADLST サイズ（概略）

種類	NPAS_POU 名称	概略 ADLST サイズ（Byte）
連続制御	NPAS_PVI	8、000
	NPAS_PID	17、000
	NPAS_MLD	5、000
	NPAS_MID_PB	8、000
	NPAS_RATIO_RT	13、000
	NPAS_PG_L30_BP	7、000
	NPAS_VELLIM_PB	9、000
数値演算	NPAS_FUNC_VAR	6、000
	NPAS_BDBUF_R/T	1、000
シーケンス	NPAS_SO_2	4、000
	NPAS_SIO_22	6、000

注：　プログラムインスタンス名は 4 文字、POU インスタンス名は 6 文字として算出

表　OPC チェック有り変数の ADLST サイズ（概略）

種類	変数のデータ型	概略 ADLST サイズ（Byte）
OPC のチェックの有る変数	BOOL	160
	REAL	160
	STRING	160
	CData_REAL	160
	Dtag_I_Anlg/_O_Anlg	160
	Dtag_I_Sts/_O_sts	160

注：プログラムインスタンス名は 4 文字、変数名は 6 文字として算出

計算例：

NPAS_PVI×100 個、NPAS_PID×50 個、OPC のチェックが有る変数×20 個の場合

ADLST サイズ＝
17、000＋8、000×100＋17、000×50＋160×20＝1、670、200Byte ＜ 4MByte

● リテインデータ容量の算出

リテインデータ容量は下記の式と下表を用いて計算します。

リテインデータ容量＝NPAS_POU のリテインデータサイズ合計
＋保持変数のチェックが有る変数のリテインデータサイズ合計

表　NPAS POU のリテインデータ容量

種類	NPAS_POU 名称	概略リテインデータサイズ（Byte）
連続制御	NPAS_PVI	350
	NPAS_PID	800
	NPAS_MLD	250
	NPAS_MID_PB	350
	NPAS_RATIO_RT	650
	NPAS_PG_L30_BP	500
	NPAS_VELLIM_PB	450
数値演算	NPAS_FUNC_VAR	550
	NPAS_BDBUF_R/T	450
シーケンス	NPAS_SO_2	200
	NPAS_SIO_22	250

注：プログラムインスタンス名は 4 文字、POU インスタンス名は 6 文字として算出

表　保持変数のリテインデータ容量

種類	変数のデータ型	リテインデータサイズ（Byte）
保持変数のチェックが有る変数	BOOL	1
	REAL	4
	STRING	86
	CData_REAL	32

注：プログラムインスタンス名は 4 文字、変数名は 6 文字として算出

計算例：

NPAS_PVI×100 個、NPAS_PID×50 個、保持変数のチェックが有る CData_REAL 変数×20 個の場合

リテインデータ容量＝250×100＋800×50＋32×20＝75、640Byte＜360KByte

B4.5.3 パフォーマンスの確認

制御ロジックの実行時間の目安を示し、この値から制御周期を算出する方法を説明します。また、制御周期の要求が優先される場合の最大ループ数の目安を示します。

■ NPAS_POU の実行時間と制御周期

制御内容がある程度具体的になっている場合のパフォーマンス確認方法を説明します。

● 条件

・ 保持変数指定：有効（ハードバックアップ有り）

● 代表的な POU の実行時間

表　代表的な POU の実行時間

POU 名	実行時間
NPAS_PID	約 1.1 ms
NPAS_PVI	約 0.4 ms

● その他の NPAS_POU の実行時間

下表に従って、NPAS_PID または NPAS_PVI に置き換えて、換算してください。

表　パフォーマンスの目安（NPAS_POU 別）

分類	NPAS_PID 相当	NPAS_PVI 相当
連続制御	NPAS_PID	NPAS_PVI
	NPAS_PI_HLD	NPAS_MLD
	NPAS_ONOFF	NPAS_MLD_PB
	NPAS_ONOFF_G	NPAS_MLD_BT
	NPAS_RATIO	NPAS_AS_H/M/L
	NPAS_RATIO_RT	
	NPAS_PG_L30	
	NPAS_PG_L30_BP	
	NPAS_VELLIM	
	NPAS_VELLIM_PB	
数値演算	NPAS_FUNC_VAR	NPAS_FOUT
	NPAS_TP_CFL	NPAS_LDLAG
		NPAS_DLAY
		NPAS_AVE_M/C
		NPAS_T_CFL
		NPAS_P_CFL
		NPAS_SW13/SW31
		NPAS_SW19/SW91
		NPAS_ASTM1/2
		NPAS_BDBUF_R/T
シーケンス制御		NPAS_SIO*
		NPAS_TM
		NPAS_CT

● POU 個数と制御周期の目安

制御周期は、下記の計算式でその目安を算出できます。

$$\text{制御周期（ms）} > \frac{\text{（NPAS_PID 相当 POU 個数}\times 1.1\text{ ms）＋（NPAS_PVI 相当 POU 個数}\times 0.4\text{ ms）}}{\text{最大 CPU 負荷率}}$$

注： 最大 CPU 負荷率=0.6（60%）

■ 制御周期と最大ループ数の目安

制御周期の要求が優先される場合の最大ループ数の目安を下表に示します。

● 条件

- ループ構成：アナログ入力（1 点）、アナログ出力（1 点）、NPAS_PID（1 個）
- 保持変数指定：有効、制御周期：スキャン周期と同一、CPU スタイル：S3

● 最大ループ数の目安

表 最大ループ数の目安

制御周期	最大ループ数（推奨値）	
（ms）	シングル CPU	二重化 CPU
20	8	4
50	24	20
100	50	46
200	103	97

B4.5.4　入出力処理のパフォーマンス

■ 入出力応答時間とは

入力信号を入力 I/O モジュールで取り込み、これを用いて CPU で制御演算処理を実行し、さらに出力 I/O モジュールから信号出力を実行するまでの時間を「入出力応答時間」と呼びます。ここでは、この入出力応答時間の見積方法を示します。

■ 入出力応答時間の内訳

入出力応答時間は、下記 3 つの処理時間の和となります。

Ti：　入力 I/O モジュールにおける入力処理時間

Tcpu ：FCN/FCJ 制御機能の処理に要する時間

To：　出力 I/O モジュールにおける出力処理時間

参照

FCN/FCJ 制御機能の実行周期については、「B4.5.1 FCN/FCJ のスキャン周期と制御周期」を参照してください。

■ 動作とタイミング

I/O モジュールとして、アナログ入力モジュール、アナログ出力モジュールを用い、PID などの制御動作を実行する場合を例にとった事例を下図に示します。入力信号の変化はスキャン周期と非同期のため、Tcpu は Tcpu (min)から Tcpu (max)まで変動します。

Tcpu（min）：入力信号の変化直後に CPU へ読込が実行された場合

Tcpu（max）：入力信号の変化直前に CPU への読込が実行され、信号変化が 1 制御周期遅れて認知された場合

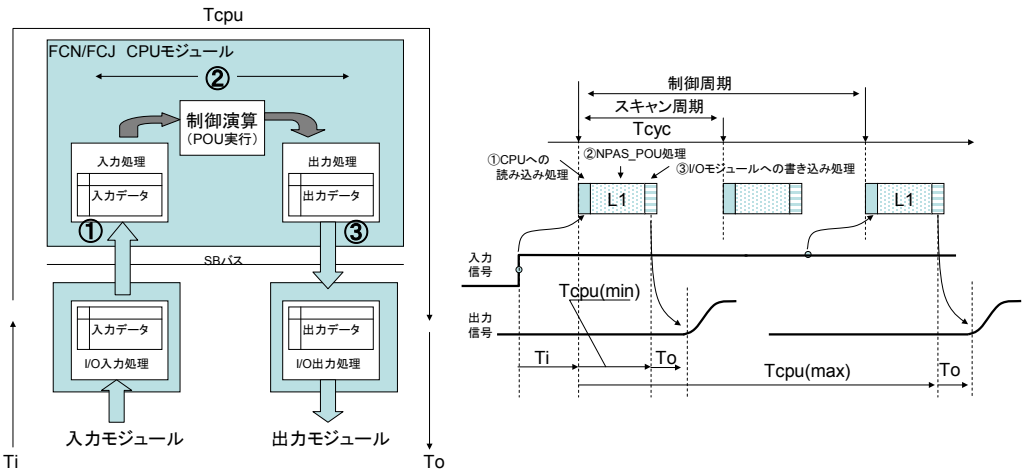


図　入力／演算／出力の仕組みと入出力タイミング

参照

PID 等の制御演算は制御周期ごとに実行されます。制御周期はスキャン周期の整数倍の周期を設定します。制御周期とスキャン周期の関係については、「B4.5.1 FCN/FCJ のスキャン周期と制御周期」を参照してください。

■ 入出力応答時間の目安（PID 制御などの場合）

Tcpu は制御周期の時間分だけ応答速度が変化します。このため、以下の簡易計算式を用いて目安を算出してください。

● 計算式（制御周期 50ms 以上の場合）

【条件】アナログ入出力モジュール：NFAB841、シングル CPU（S3）

入出力応答時間＝Ti+To+Tcpu

入出力応答時間（min）＝25ms＋スキャン周期×0.6

入出力応答時間（max）＝25ms＋スキャン周期×0.6＋制御周期

● 計算例

【条件】スキャン周期；100 ms、制御周期：100 ms

入出力応答時間（min）＝25+100×0.6=85 ms

入出力応答時間（max）＝25+100×0.6＋100=185 ms

補足

I/O モジュールの実装数は、入出力応答時間に影響しません。

B5. 製品一覧

B5.1 ソフトウェア製品一覧

FCN/FCJ 向けのソフトウェア製品は、大別して各種プログラムが格納されたソフトウェア媒体と、それらプログラムを使用（動作）可能にするためのライセンス ID があります。

● ソフトウェア媒体

制御アプリケーションを開発する汎用パソコンに、DVD-ROM で提供する各種ソフトウェアプログラムをインストールします。

- FCN/FCJ ソフトウェア媒体
  ロジックデザイナ／リソースコンフィギュレータ／PAS ポートフォリオプログラム、FCN/FCJ OPC サーバ for Windows および FCN/FCJ 電子ドキュメント（取扱説明書）
- アプリケーションポートフォリオソフトウェア媒体
  アプリケーションポートフォリオのうち PAS ポートフォリオを除く全種類の実行プログラム、および電子ドキュメント（取扱説明書）

表　ソフトウェア媒体（DVD-ROM）

製品名	形名	基本仕様コード	備考
FCN/FCJ ソフトウェア媒体	NT203AJ	PC11A	－
FCN/FCJ APPF（アプリケーションポートフォリオ）ソフトウェア媒体	NT205AJ	PC11A	－

● ソフトウェアライセンス

ソフトウェア媒体で提供される各種プログラムを使用（動作）可能にするために、オーダ ID シートと呼ばれる紙面で提供されます。このオーダ ID シートには、オーダ ID とパスワードが記載されています。購入後、当社の Web サイトにアクセスしてオーダ ID とパスワードを入力すると、それぞれのソフトウェアに対応するライセンスが発行されます。
ライセンスには、ライセンス ID とキーコードの 2 つがあります。

表　FCN/FCJ 基本ライセンス

製品名	形名	基本仕様コード		
		システムカードサイズ／Java 機能		
		128MB／無し	128MB／有り	512MB／有り
FCN/FCJ 基本ソフトウェアライセンス　シングル用（システムカード付属）	NT711AJ	LS05A	LM05A	LM04A
FCN 基本ソフトウェアライセンス　二重化用（システムカード 128MB 二枚付属）	NT712AJ	LM05A	—	—

表　FCN/FCJ Java 機能実行ライセンス

製品名	形名	基本仕様コード
FCN/FCJ　Java 機能実行ライセンス	NT719AJ	LW11A

注：　NT711AJ-LS0□□（Java 機能なし）にのみ適用可能です。NT712AJ（二重化用）には適用できません。

表　FCN/FCJ アプリケーションポートフォリオライセンス

製品名	形名	基本仕様コード	収録ソフトウェア媒体（形名）	
			NT205AJ	NT203AJ
PAS ポートフォリオ	NT8001J	LW11A		○
SAMA ポートフォリオ	NT8002J	LW11A	○	
Webmetry 基本ライブラリポートフォリオ	NT8010J	LW11A	○	
Web アプリケーションポートフォリオ（InfoWell）	NT8012J	LW11A	○	
E-mail アプリケーションポートフォリオ（InfoWell）	NT8013J	LW11A	○	
グラフィックポートフォリオ（InfoWell）	NT8014J	LW11A	○	
ロギングポートフォリオ（InfoWell）	NT8016J	LW11A	○	
FA-M3 通信ポートフォリオ	NT8020J	LW11A	○	
MELSEC 通信ポートフォリオ	NT8021J	LW11A	○	
SYSMAC 通信ポートフォリオ	NT8022J	LW11A	○	
電力モニタ通信ポートフォリオ	NT8030J	LW11A	○	
温調計通信ポートフォリオ	NT8031J	LW11A	○	
Modbus 通信ポートフォリオ	NT8035J	LW11A	○	
DNP3 通信ポートフォリオ	NT8036J	LW11A	○	
時刻同期サーバポートフォリオ	NT8040J	LW11A		○
ボイラ制御ポートフォリオ	NT8101J	LW11A	○	
ボイラ補機制御ポートフォリオ	NT8102J	LW11A	○	

表　FCN/FCJ エンジニアリングライセンス

製品名	形名	基本仕様コード	備考
FCN/FCJ ロジックデザイナ	NT751FJ	LW11A	(*1)、(*2)
FCN/FCJ シミュレータ（USB 接続 ID モジュール）	NT752AJ	LU11A	(*1)、(*2)
FCN/FCJ　Java アプリケーション開発キット	NT755FJ	LW11A	(*1)、(*2)

*1:　収録ソフトウェア媒体は、形名：NT203AJ です。
*2:　パソコン側にライセンスが必要です。

表　FCN/FCJ OPC サーバライセンス

製品名	形名	基本仕様コード	備考
FCN/FCJ OPC サーバ for Windows	NT781AJ	LW11A	(*1)、(*2)
FCN/FCJ OPC サーバ用ネットワーク二重化	NT783AJ	LW11A	(*1)、(*2)

*1:　収録ソフトウェア媒体は、形名：NT203AJ です。
*2:　パソコン側にライセンスが必要です。

B5.2 アプリケーション開発に必要なソフトウェア

● ロジックデザイナ、FCN/FCJ Java アプリケーション開発キット

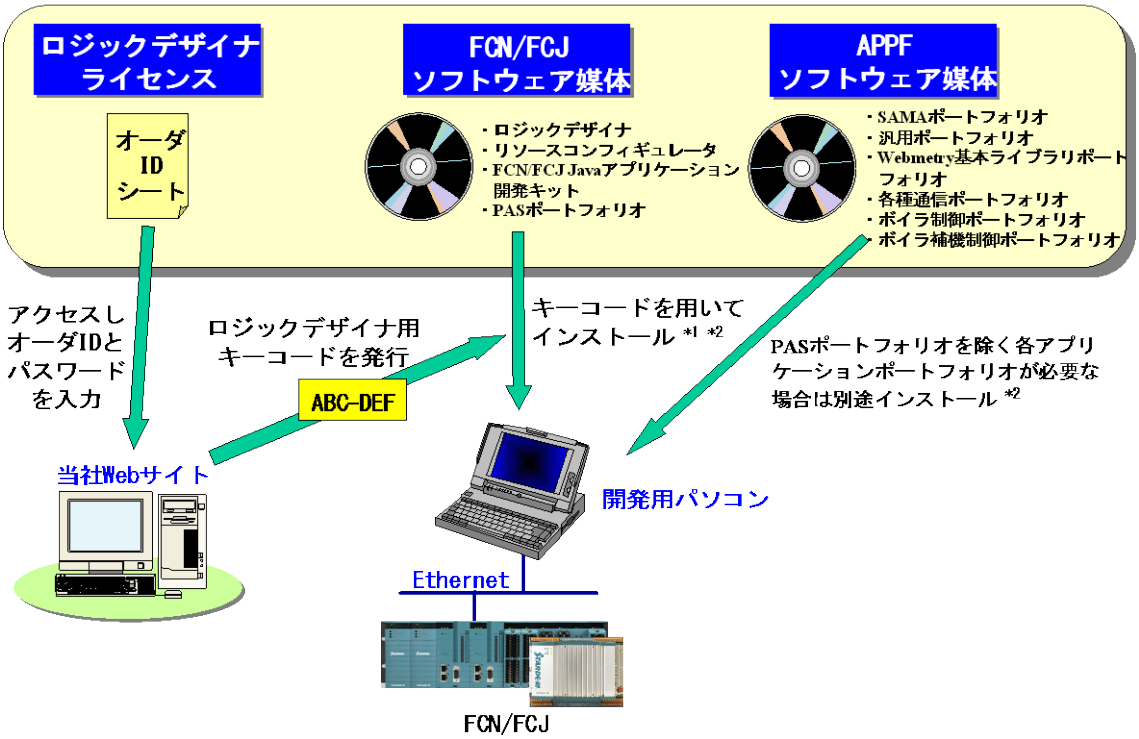
制御アプリケーション開発用パソコンにロジックデザイナまたは FCN/FCJ Java アプリケーション開発キットをインストールするには、FCN/FCJ ソフトウェア媒体（DVD-ROM）のほかにそれらを動作させるためのライセンスが必要です。
各ライセンスは、オーダ ID シートと呼ばれる紙面で提供され、そこに記載されているオーダ ID とパスワードを基に当社 Web サイトにアクセスすると、キーコードが発行されます。このキーコードを使用すればロジックデザイナまたは FCN/FCJ Java アプリケーション開発キットを開発用パソコンにインストールして使用できます。

● リソースコンフィギュレータ

リソースコンフィギュレータはライセンス不要です。FCN/FCJ ソフトウェア媒体（DVD-ROM）からリソースコンフィギュレータを開発用パソコンにインストールし使用できます。

● アプリケーションポートフォリオ

各アプリケーションポートフォリオも上記リソースコンフィギュレータ同様、ライセンス無しで開発用パソコンにインストールし使用できます。ただし、アプリケーションポートフォリオを使用したアプリケーションを FCN/FCJ にダウンロードし実行させる際、対象の FCN/FCJ に動作させたいアプリケーションポートフォリオのライセンスを登録しておく必要があります。



*1: リソースコンフィギュレータは、キーコード無しでインストールし使用可能です。
*2: 各アプリケーションポートフォリオは、キーコード無しでインストールし使用可能ですが、FCN/FCJ 上で動作させるためには別途ライセンスが必要です。

図 アプリケーション開発に必要なソフトウェア

B5.3 FCN/FCJ 動作に必要なソフトウェア

制御アプリケーションを FCN/FCJ 上で動作させるために、以下の各ライセンスをあらかじめ FCN/FCJ に登録しておく必要があります。各ライセンスは、オーダ ID シートと呼ばれる紙面で提供され、そこに記載されているオーダ ID とパスワードを基に当社 Web サイトにアクセスし、ライセンス ID（ファイル）を取得します。リソースコンフィギュレータを用いてこのライセンス ID を FCN/FCJ に登録すれば該当の機能を動作させられます。

● FCN/FCJ 基本ソフトウェアライセンス

FCN/FCJ を動作させるためのソフトウェアライセンスです。FCN/FCJ に実装する「FCN/FCJ システムカード」とオーダ ID シートで提供されます（FCN 二重化用の基本ソフトライセンスは、システムカードを 2 枚提供）。システムカードには、リアルタイム OS や制御実行エンジンなどが書き込まれています。ここに、アプリケーションポートフォリオライセンスなどをあらかじめ登録しておくと、これらを使用した制御アプリケーションが動作可能となります。

● アプリケーションポートフォリオライセンス

アプリケーションポートフォリオを用いた制御アプリケーションを FCN/FCJ で動作させる場合、あらかじめ対象の FCN/FCJ に使用しているアプリケーションポートフォリオのライセンスを登録しておく必要があります。

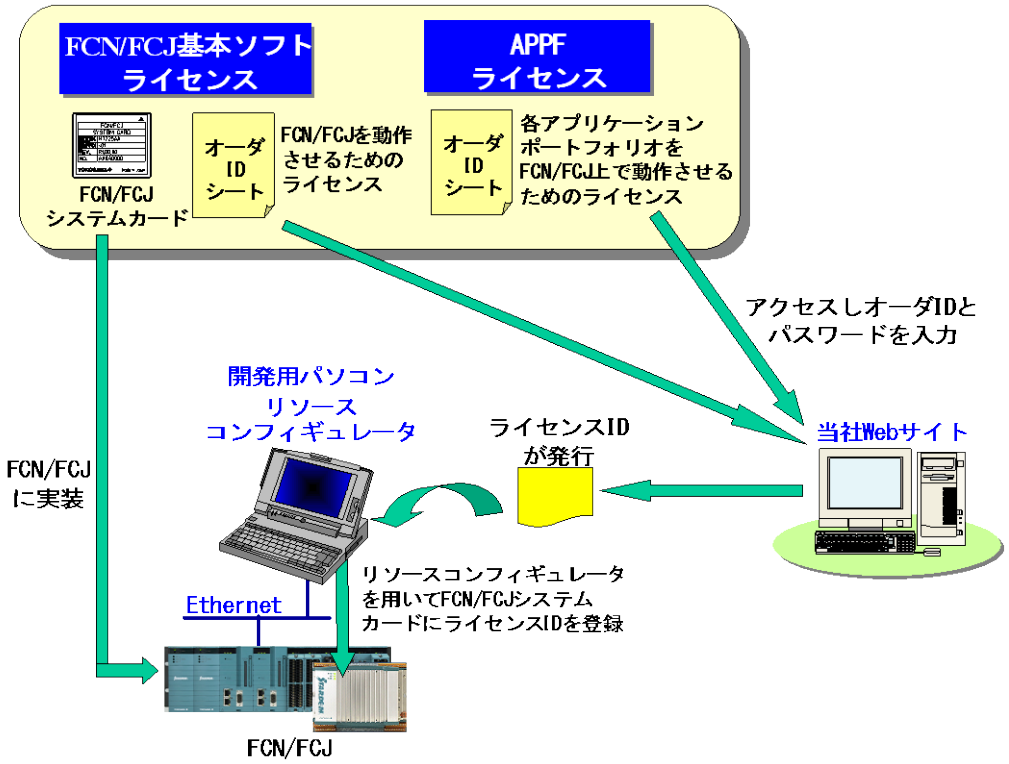


図　アプリケーション開発に必要なソフトウェア

B5.4 ハードウェア製品一覧

FCN/FCJ のハードウェアは、下表のとおりです。

表 FCN/FCJ ハードウェア一覧（1/2）

	製品形名	製品名称
FCJ 本体	NFJT100	自律型コントローラ FCJ
FCN 共通部	NFBU200	FCN 用ベースモジュール
	NFBU050	ショートベースモジュール
	NFPW44□	FCN 用電源モジュール
	NFCP100	FCN 用 CPU モジュール(*1)
	NFSB100	FCN 用 SB バスリピートモジュール(*2)
	NFCB301	SB バスケーブル
FCN アナログ I/O モジュール	NFAI135	アナログ入力モジュール（4～20mA、8 点、個別絶縁）
	NFAP135	パルス入力モジュール（8 点、0～10kHz、個別絶縁）
	NFAF135	周波数入力モジュール（8 点、0.1Hz～10kHz、個別絶縁）
	AAT145	熱電対／mV 入力モジュール（16 点、個別絶縁）
	AAR145	測温抵抗体／すべり抵抗入力モジュール（16 点、個別絶縁）
	NFAI141	アナログ入力モジュール（4～20mA、16 点、非絶縁）
	NFAV141	アナログ入力モジュール（1～5V、16 点、非絶縁）
	NFAV142	アナログ入力モジュール（-10～+10V、16 点、非絶縁）
	NFAT141	熱電対／mV 入力モジュール（16 点、一括絶縁）
	NFAI143	アナログ入力モジュール（4～20mA、16 点、一括絶縁）
	NFAV144	アナログ入力モジュール（-10～+10V、16 点、一括絶縁）
	NFAR181	測温抵抗体入力モジュール（12 点、一括絶縁）
	AAP149	パルス列入力モジュール（PM1 互換モジュール） （16 点、パルス数カウント、0～6kHz、非絶縁）
	NFAI835	アナログ入出力モジュール （4～20mA／4 点入力、4～20mA／4 点出力、個別絶縁）
	NFAI841	アナログ入出力モジュール （4～20mA／8 点入力、4～20mA／8 点出力、非絶縁）
	NFAB841	アナログ入出力モジュール （1～5V／8 点入力、4～20mA／8 点出力、非絶縁）
	NFAV542	アナログ出力モジュール（-10～+10V、16 点、非絶縁）
	NFAI543	アナログ出力モジュール（4～20mA、16 点、一括絶縁）
	NFAV544	アナログ出力モジュール（-10～+10V、16 点、一括絶縁）
FCN デジタル I/O モジュール	NFDV151	デジタル入力モジュール（32 点、24VDC、一括絶縁）
	NFDV141	デジタル入力モジュール（16 点、100～120VAC、一括絶縁）
	NFDV142	デジタル入力モジュール（16 点、200～240VAC、一括絶縁）
	NFDV157	デジタル入力モジュール（32 点、24VDC、一括絶縁、押し締め端子専用）
	NFDV161	デジタル入力モジュール（64 点、24VDC、一括絶縁）
	NFDV551	デジタル出力モジュール（32 点、24VDC、一括絶縁）
	NFDR541	リレー出力モジュール（16 点、24～110VDC／100～240VAC、一括絶縁）
	NFDV557	デジタル出力モジュール（32 点、24VDC、一括絶縁、押し締め端子専用）
	NFDV561	デジタル出力モジュール（64 点、24VDC、一括絶縁）
	NFDV532	パルス幅出力モジュール （4 チャネル：Up パルス／Down パルス各 1 点、24VDC、一括絶縁）

*1: NFCP100 は NFBU050 上で二重化できません。
*2: NFSB100 は NFBU050 上で使用できません。

表　FCN/FCJ ハードウェア一覧（2/2）

	製品形名	製品名称
FCN 通信モジュール	NFLF111	Foundation fieldbus 通信モジュール
	NFLR111	RS-232-C 通信モジュール（2 ポート、300bps～115.2kbps）
	NFLR121	RS-422／RS-485 通信モジュール（2 ポート、300bps～115.2kbps）
	NFLC121	CANopen 通信モジュール（1 ポート、10kbps～1Mbps）
	NFLP121	PROFIBUS-DP 通信モジュール（1 ポート、9.6kbps～12Mbps）
FCN ST カード互換用デジタル I/O モジュール	ADV159	デジタル入力モジュール（ST3／PB5 互換モジュール） （32 点入力、各点絶縁）
	ADV559	デジタル出力モジュール（ST4 互換モジュール） （32 点出力、各点絶縁）
	ADV859	デジタル入出力モジュール（ST2 互換モジュール） （16 点入力／16 点出力、各点絶縁）
	ADV169	デジタル入力モジュール（ST6 互換モジュール） （64 点入力、一括絶縁、16 点コモン共通）
	ADV569	デジタル出力モジュール（ST7 互換モジュール） （64 点出力、一括絶縁、16 点コモン共通）
	ADV869	デジタル入出力モジュール（ST5 互換モジュール） （32 点入力／32 点出力、一括絶縁、16 点コモン共通）
FCN カバー	NFDCV01	ダミーカバー（I/O モジュール用）
	NFDCV02	ダミーカバー（FCN 電源モジュール用）
	NFCCC01	MIL ケーブルコネクタ用カバー
FCN SB バス分岐栓	NFSBT01	SB バス用分岐栓
	NFSBT02	SB バス用ターミネータ内蔵形分岐栓
FCN 押し締め端子ブロック	NFTA4S	アナログ用押し締め端子ブロック（16 点）
	NFTT4S	熱電対／mV 用押し締め端子ブロック（16 点）
	NFTR8S	測温抵抗体／すべり抵抗用押し締め端子ブロック（12 点）
	NFTB5S	デジタル入力用押し締め端子ブロック（32 点）
	NFTD5S	デジタル出力用押し締め端子ブロック（32 点）
	NFTI3S	アナログ個別絶縁用押し締め端子ブロック （NFAI135、NFAP135、NFAF135：8 点、NFAI835：入力 4 点／出力 4 点）
	NFTC4S	デジタル用押し締め端子ブロック （16 点、専用コネクタ、サージアブソーバなし）
	NFTC5S	デジタル用押し締め端子ブロック（32 点、専用コネクタ）
	NFTF9S	フィールドバス用押し締め端子ブロック
FCN コネクタ端子台	TAS40	MIL コネクタ端子台（アナログ入出力モジュール用）
	TAS50	MIL コネクタ端子台 （NFDV151、NFDV161、NFDV532、NFDV551、NFDV561 用）
ケーブル	KMS40	MIL コネクタケーブル（アナログ入出力モジュール用）
	KMS50	MIL コネクタケーブル （NFDV151、NFDV161、NFDV532、NFDV551、NFDV561 用）
	KMSG7	リレーターミナルケーブル （NFDV151、NFDV161、NFDV532、NFDV551、NFDV561 用）
KS ケーブルインタフェースアダプタ	ATK4A	KS ケーブルインタフェースアダプタ（アナログ 16 点用）
	ATI3A	KS ケーブルインタフェースアダプタ（アナログ 8 点用）
	ATB3A	KS ケーブルインタフェースアダプタ（アナログ入力 4 点／出力 4 点用）
	ATD5A	KS ケーブルインタフェースアダプタ（デジタル 32 点用）
	ATM4A	KS ケーブルインタフェースアダプタ（MAC2 互換用）
	ATV4A	KS ケーブルインタフェースアダプタ（VM2 互換用）

参照

- FCN/FCJ のハードウェアについては、「B3. FCN/FCJ ハードウェア」を参照してください。
- 各モジュールの機能の詳細については、「F2. GS 集：FCN/FCJ 関連」を参照してください。

B6.　保守

B6.1　全データのバックアップ／リストア（R3.30 以降）

FCN/FCJ システムカードの故障など、システム異常の発生に備えて、定期的に FCN/FCJ システムカードのデータをバックアップしてください。
本節では、FCN/FCJ のシステムカードのファイルの全データのバックアップ／リストア機能を説明します。

B6.1.1　全データのバックアップ

以下の手順で、バックアップを行います。

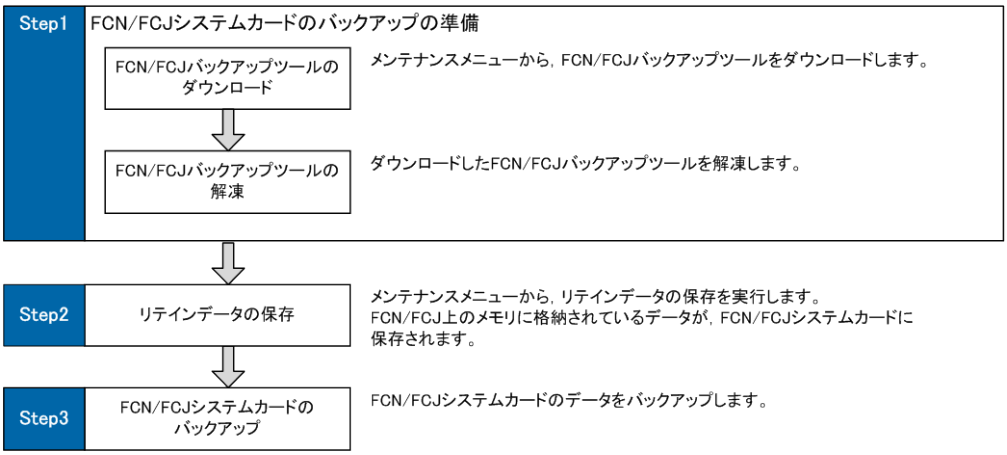


図　全データのバックアップ手順

重　要

ロギングデータは、全データのバックアップ対象外です。バックアップの詳細は、各ロギングツールの説明書を参照してください。

■ Step 1: FCN/FCJ システムカードのバックアップの準備

● FCN/FCJ バックアップツールのダウンロード

1. Web ブラウザを起動し、「STARDOM Maintenance Page」を開きます。
2. ［Maintenance Menu］を選択します。
   「FCX Maintenance Menu」が表示されます。
3. ［Download the Backup Tools］を選択します。
   FCN/FCJ バックアップツール（FCXTOOL.EXE）がダウンロードされます。

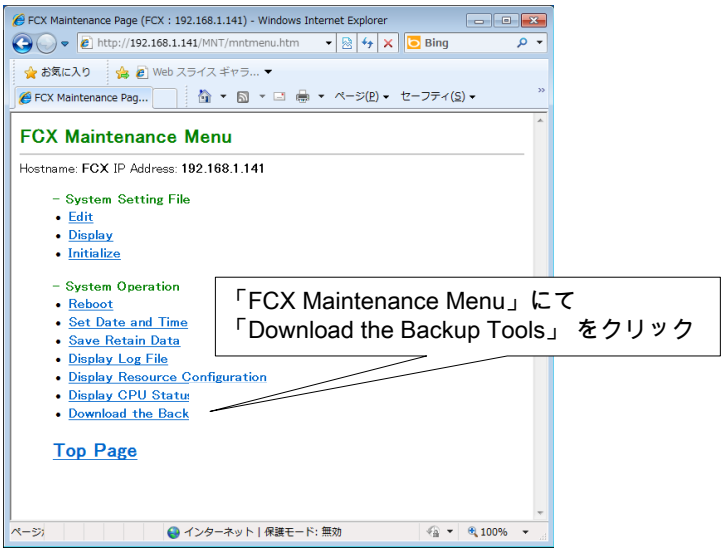


図　FCN/FCJ バックアップツールのダウンロード

● FCN/FCJ バックアップツールの解凍

1. FCN/FCJ バックアップツール（FCXTOOL.EXE）をダブルクリックします。
2. 解凍先フォルダ名を指定して、［Install］ボタンを押します。
   FCN/FCJ バックアップツールが解凍されます。

補足

FCN/FCJ バックアップツールの解凍先フォルダに「C:¥temp」を指定した場合、以下のように解凍されます。

```
C:¥temp¥FcxTool¥FcxBackup.exe
C:¥temp¥FcxTool¥FcxRestore.exe
C:¥temp¥FcxTool¥backuplist.txt
C:¥temp¥FcxTool¥backuplist_all.txt
```

重 要

解凍先のフォルダは、スペースを含んだディレクトリ（例：「C: ¥Documents and Settings¥Default User¥My Documents」）を指定しないでください。
スペースを含むディレクトリ上のコマンドを実行すると、バックアップリストアのコマンドが、途中でエラーにより停止します。

■ Step 2: リテインデータの保存

1. Web ブラウザを起動し、「STARDOM Maintenance Page」を開きます。
2. ［Maintenance Menu］を選択します。
   「FCX Maintenance Menu」が表示されます。
3. ［Save Retain Data］を選択します。
   現時点で保存されているリテインデータの保存日時が表示されます。
4. ［SAVE］ボタンを押します。
   リテインデータが保存されます。

補足

メンテナンスモードの場合は、［SAVE］ボタンは表示されません。

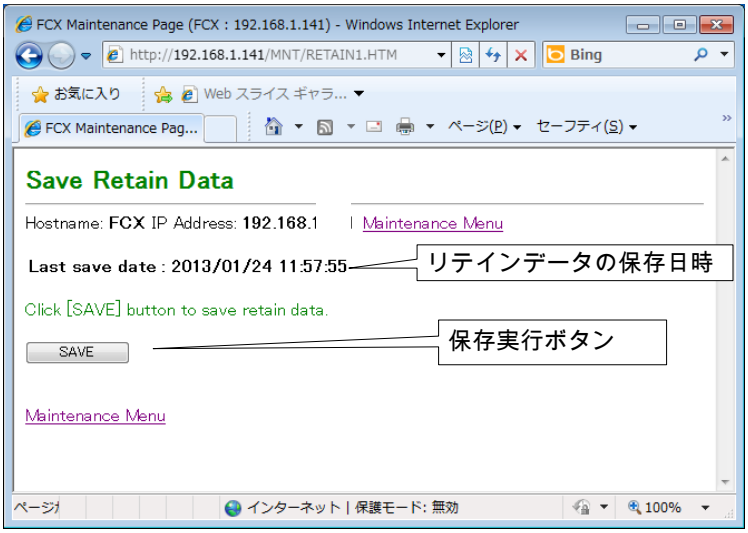


図　リテインデータの保存

参照

Web ブラウザの操作方法は、「B6.2 メンテナンスメニュー」を参照してください。

■ Step 3: FCN/FCJ システムカードのバックアップ

1. コマンドプロンプトを起動します。
2. コマンドプロンプト上で、カレントディレクトリを FCN/FCJ バックアップツールを解凍したフォルダに移動します。
   FCN/FCJ バックアップツールの解凍先フォルダに「C:¥temp」を指定した場合「cd C:¥temp¥FcxTool」と入力し、［Enter］キーを押します。
3. 「FcxBackup -all ＜ホスト名 or IP アドレス＞」を入力し、［Enter］キーを押します。
   カレントディレクトリに「BACKUP」フォルダが作成されます。

補足

- FCN/FCJ をメンテナンスモードにしてシステム設定ファイルのバックアップを行う場合は、「FCX Maintenance Menu」から「STARDOM Maintenance Page」に切り替えてください。
  「FCX Maintenance Menu」を開いたままの場合や、開いたまま Web ブラウザを終了してしまった場合はバックアップが行えません。
  「FCX Maintenance Menu」を開いたまま Web ブラウザを終了した場合は、再度 Web ブラウザを起動し、「STARDOM Maintenance Page」から「FCX Maintenance Menu」を表示した後、「STARDOM Maintenance Page」に戻って、Web ブラウザを終了してください。
- FCN/FCJ システムカードのバックアップには、15 分程度かかります。
- 「FcxBackup」コマンドで「-all」指定のありなしでは、バックアップ対象が異なります。必ず「-all」を指定をしてください。

バックアップ対象	「-all」指定あり	「-all」指定なし
IEC アプリケーション	○	○
コンフィギュレーション (IO、IP アドレスなど)	○	○
リテインセーブのデータ	○	○
基本ソフト	○	×
Java アプリケーション	○	×
ロギングデータ	×	×
トレンドデータ	×	×
レポートデータ	×	×

重 要

バックアップに失敗した場合、以下の点を確認してください。

- FCN/FCJ のホスト名あるいは IP アドレスに間違いはないか。
  Ping コマンドを使用してチェックしてください。
- スペースを含んだディレクトリ下にあるコマンドを実行していないか。
  スペースを含まないディレクトリ下にコマンドを移動してください。
- 上記で解決しない場合は、「FcxTool¥TEMP¥FTPLOG.TXT」にバックアップ操作のログが記録されますので内容をチェックしてください。

B6.1.2 全データのリストア

FCN/FCJ システムカードを交換しない場合は、下図の Step 3 以降を実施してください。FCN/FCJ システムカードを交換する場合は、下図の Step1 以降を実施してください。

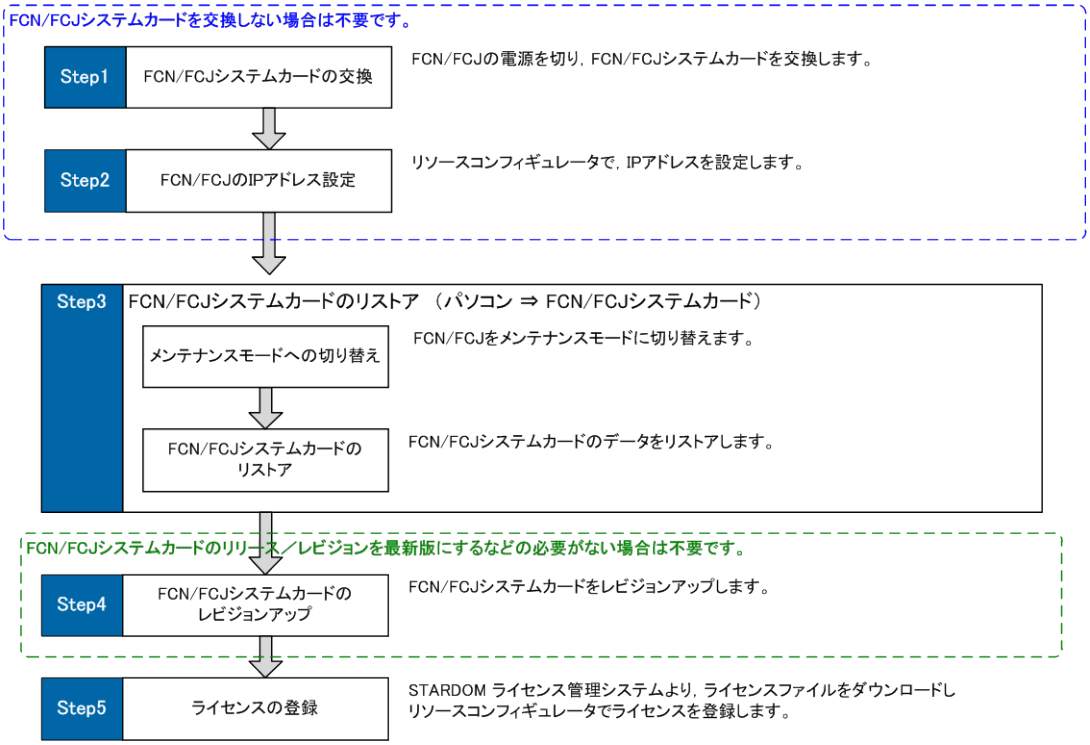


図　全データのリストア手順

■ Step 1: FCN/FCJ システムカードの交換

FCN/FCJ の電源を切り、FCN/FCJ システムカードを交換します。

補足

FCN/FCJ システムカードを交換しない場合は、本ステップ（FCN/FCJ システムカードの交換）は不要です。

■ Step 2: FCN/FCJ の IP アドレス設定

リソースコンフィギュレータで、FCN/FCJ の IP アドレスを設定します。

補足

- FCN/FCJ システムカードを交換しない場合は、本ステップ（FCN/FCJ の IP アドレス設定）は不要です。
- FCN/FCJ の IP アドレス設定手順は、リソースコンフィギュレータのヘルプを参照してください。

■ Step 3: FCN/FCJ システムカードのリストア

● メンテナンスモードへの切り替え

1. Web ブラウザを起動し、「STARDOM Maintenance Page」を開きます。
2. ［Maintenance Menu］を選択します。
   「FCX Maintenance Menu」が表示されます。
3. ［Reboot］を選択します。
   「Reboot Menu」が表示されます。
4. ［Reboot（Maintenance Mode）］を選択し、［OK］ボタンを押します。
   FCN/FCJ がメンテナンスモードで立ち上がります。

● FCN/FCJ システムカードのリストア

1. コマンドプロンプトを起動します。
2. コマンドプロンプト上で、カレントディレクトリを FCN/FCJ バックアップツールを解凍したフォルダに移動します。
   FCN/FCJ バックアップツールの解凍先フォルダに「C:¥temp」を指定した場合「cd C:¥temp¥FcxTool」と入力し、［Enter］キーを押します。
3. 「FcxRestore ＜ホスト名 or IP アドレス＞」を入力し、［Enter］キーを押します。

補足

- リストアを実行するパソコンとバックアップを実施したパソコンが異なる場合は、手順 1（コマンドプロンプトの起動）の前にバックアップした「Backup」フォルダ以下のファイルを、リストアを実行するパソコンの FCN/FCJ バックアップツールを解凍したフォルダにコピーしてください。
- バックアップしたシステム設定ファイルを別の FCN/FCJ システムカードにリストアした場合は、リストアしたライセンスファイルが FCN/FCJ システムカードに対応していないため、そのままでは動作しません。リソースコンフィギュレータを使って、その FCN/FCJ システムカードに対応したライセンスファイルを上書きすることで動作するようになります。

重 要

リストアに使用したシステムカードは、バックアップ元のシステムのリリースになります。

■ Step 4: FCN/FCJ システムカードのレビジョンアップ

FCN/FCJ システムカードのリリース／レビジョンを最新にアップグレードします。

補足

FCN/FCJ システムカードのリリース／レビジョンを最新版にする必要がない場合、本ステップは不要です。

重 要

FCN/FCJ システムカードのレビジョンアップを行うためには、フラッシュメモリに約 10Mbyte の空き領域が必要です。「STARDOM Maintenance Page」の［Flash Memory (FREE)］を参照し、空き領域を確認してください。空き領域が不足している場合は、不要なファイルを削除してください。

参照

FCN/FCJ システムカードのレビジョンアップ／レビジョンダウンの詳細は、FCN/FCJ 基本ソフトウェア リリースノートを参照してください。

● FCN/FCJ 基本ソフトウェアパッケージの取得

1. 変更するリリース／レビジョンの DVD-ROM をドライブに挿入します。
2. ［DVD 内容］を選択します。
3. FCN/FCJ 基本ソフトウェアパッケージ「Pkg_FCX」フォルダ内の Pkg_FCX.exe をダブルクリックします。
4. 解凍先フォルダ名を指定して、［Install］ボタンを押します。
   FCN/FCJ 基本ソフトウェアパッケージが解凍されます。

補足

FCN/FCJ 基本ソフトウェアパッケージの解凍先フォルダに「C:¥temp」を指定した場合、以下のように解凍されます。

```
C:¥temp¥FCX¥FcxRevup.exe
```

● メンテナンスモードへの切り替え

1. Web ブラウザを起動し、「STARDOM Maintenance Page」を開きます。
2. ［Maintenance Menu］を選択します。
   「FCX Maintenance Menu」が表示されます。
3. ［Reboot］を選択します。
   「Reboot Menu」が表示されます。
4. ［Reboot（Maintenance Mode）］を選択し、［OK］ボタンを押します。
   FCN/FCJ がメンテナンスモードで立ち上がります。

● FCN/FCJ システムカードのレビジョンアップ

1. コマンドプロンプトを起動します。
2. コマンドプロンプト上で、カレントディレクトリを FCN/FCJ 基本ソフトウェアパッケージを解凍したフォルダに移動します。
   FCN/FCJ 基本ソフトウェアパッケージの解凍先フォルダに「C:¥temp」を指定した場合は「cd C:¥temp¥FCX」と入力し、［Enter］キーを押します。
3. 「FcxRevup ＜ホスト名 or IP アドレス＞」を入力し、［Enter］キーを押します。

補足

レビジョンアップには、5 分程度かかります。

■ Step 5: ライセンスの登録

1. ライセンス管理システムで、ライセンスを再発行します。

補足

FCN/FCJ システムカードを交換すると、FCN/FCJ システムカードのシリアル番号が変わるためライセンスの再発行（本ステップの手順 1）が必要になります。
同じ FCN/FCJ システムカードの場合は、ライセンスの再発行は不要です。

2. ライセンス管理システムから、ライセンスファイルをダウンロードします。
3. リソースコンフィギュレータでライセンスを登録します。

参照

- STARDOM ライセンス管理システムのユーザ登録の手順は、ログインページの「初めてこのシステムをご利用される方は、こちらのページをご覧ください。」の［こちらのページ］をクリックして展開するページを参照してください。
- STARDOM ライセンス管理システムの操作方法は、ログイン後のメニューページの＜操作手順＞ボタンで表示される Help を参照してください。
- ライセンスの登録手順は、リソースコンフィギュレータのヘルプを参照してください。

B6.2 メンテナンスメニュー

メンテナンスホームページから「Maintenance Menu」を選択するとメンテナンスメニューが表示されます。
このウィンドウから以下の機能を呼び出せます。
ただし、FCN/FCJ の動作モードにより操作できる範囲に制限があります。

表　メンテナンスメニューの操作可能な機能

操作項目	動作状態	
	メンテナンス状態	オンライン状態
共通の操作 WWW サーバへの接続 メンテンナンス操作の開始・終了	○（＊1、 2）	○（＊1、 2）
システム設定ファイルの内容変更 編集 初期化	○（＊1）	×
システム設定ファイルの表示 システム設定ファイルの表示 カーネル／システムログファイルの表示	○（＊1）	○（＊3）
日付と時刻の設定	○（＊1）	○（＊4）
リブート・シャットダウン	○（＊1）	○（＊4）
リテインデータの保存	×	○（＊4）
その他 I/O コンフィグレーションの表示 CPU 動作状態の表示	○（＊1）	○（＊3）

○：　操作可能
×：　操作不可
*1:　JEROS 基本設定ファイルの“メンテナンスセキュリティ”が“YES”に設定されている場合、操作時に/JEROS ディレクトリにアクセス可能なユーザ名とパスワードの入力が必要です。
*2:　操作時に HTML ファイルの置かれている/JEROS/SYS/MNT/HTML にアクセス可能なユーザ名とパスワードの入力が必要です。
*3:　表示したいファイルにアクセス可能なユーザ名とパスワードの入力が必要です。
*4:　操作時に/JEROS ディレクトリにアクセス可能なユーザ名とパスワードの入力が必要です。

重　要

FCN/FCJ のメンテナンスには、それぞれのアクセス権を持った、ユーザ名とパスワードの指定が必要です。
アクセス権は、個別に変更できます。

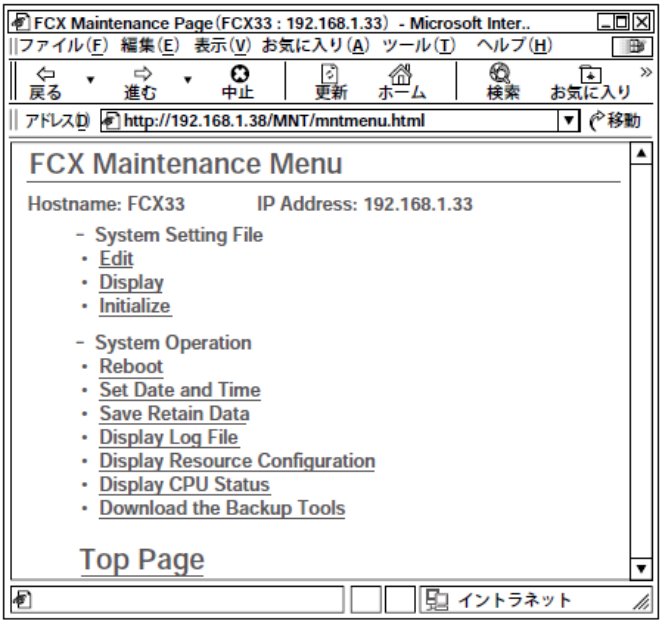


図　メンテナンスメニュー

B6.2.1　システム設定ファイル

FCN/FCJ は FCN/FCJ システムカード上に、システム設定ファイルを保有しており、その内容にしたがって動作しています。
これらのファイルを編集すれば、FCN/FCJ の動作をコントロールできます。
このシステム設定ファイルを編集、表示できます。さらに、出荷時の状態にも戻せます。
システム設定ファイルには、以下があります。

表　システム設定ファイル一覧

ファイル	説明
JEROS Basic Setting File	システム全体動作に関する定義
User Account File	FTP/WWW サーバ機能のユーザ登録
Duolet Environment Setting File	Duolet 実行環境の設定
E-Mail Configuration File	E メールの送受信機能の設定
Duolet Loader Configuration File	起動する Duolet の登録
HOSTS File	IP アドレスとホスト名の登録
DNS Setting File	DNS の設定（Java 機能でのみ使用可能）
SNTP Setting File	時刻同期機能の設定
IP Routing File	ネットワークのルーティング設定
PPP Setting File	PPP の設定
SLIP Setting File	SLIP の設定
COM1 Port Setting File	COM1 ポートのデフォルト通信条件設定
COM2 Port Setting File	COM2 ポートのデフォルト通信条件設定
Kernel Log Setting File	オペレーティングシステムのログ設定
Web Files Naming File	WWW サーバでのファイルの別名設定
MIME Type Setting File	WWW サーバのファイルの MIME タイプを登録

参照

各種システム設定ファイルの設定方法については、各設定画面のヘルプを参照してください。

B6.3 オンラインダウンロード

オンラインダウンロード機能では、FCN/FCJ の制御動作を継続しながら、制御アプリケーションを変更できます。
オンラインダウンロードの実行結果は、VDS/ASTMAC のシステムメッセージとして表示できます。
本章では、オンラインダウンロードについて説明します。

参照

オンラインダウンロード機能の詳細については、「STARDOM FCN/FCJ 解説書」（IM 34P02Q01-01）「B3. オンラインダウンロード」を参照してください。
稼働中のハードウェアのスタイル番号、ソフトウェアのリリース番号、および制御アプリケーション（プロジェクト）の作成条件の組み合わせにより、機能が制限されることがあります。
下記ホームページにてご確認ください。
http://STARDOM.jp
→会員サイト　Yokogawa Plus

B6.3.1 オンラインダウンロード機能概要

オンラインダウンロード実行可能な変更と実行できない変更を示します。

●オンラインダウンロードを実行可能な制御アプリケーションの変更

オンラインダウンロードを実行可能な制御アプリケーションの変更のうち、代表的なものを以下に示します。

- 論理 POU の変更（ローカル変数定義の変更、コードの変更など）
- 物理ハードウェア構成の変更（コンフィグレーションの追加／削除、リソースの追加／削除など）
- リソースの変更（デバイスラベル変数定義の変更、グローバル変数定義の変更など）
- プログラムが参照しているユーザライブラリの変更
- データ型定義の変更

●オンラインダウンロードを実行できない制御アプリケーションの変更

制御アプリケーションを以下のように変更した場合は、オンラインダウンロードを実行できません。

- タスクの追加／削除
- タスクタイプの変更（CYCLIC/SYSTEM/DEFAULT）
- タスク設定の変更（タスク周期、ウォッチドッグ時間、プライオリティ、スタックサイズ、SPG）
- 電力モニタ通信ポートフォリオ、温調計通信ポートフォリオの変更
- コンフィギュレーションの PLC タイプの変更
- リソースのプロセッサタイプの変更
- FCN/FCJ のアプリケーションポートフォリオのライセンス変更

B6.4 FCN/FCJ ハードウェアの保守

保守・交換作業上の注意事項、交換方法について説明します。

B6.4.1 FCN 保守・交換作業上注意事項

FCN の保守・交換作業時の注意事項を示します。

● 静電気に対する注意事項

- 保守部品は、保存／持ち運びの際、帯電防止袋の中に入れてください（FCN/FCJ の工場出荷時は、静電気に関する注意事項を貼り付けた帯電防止袋に入れられています）。
- 保守作業時は、1MΩの接地抵抗を介した接地線付リストストラップを使用してください。リストストラップは接地してください。
- 机上で作業する場合は、1MΩの接地抵抗を介して接地した導電性シートの上で作業してください。作業者はリストストラップを着用してください。作業台のまわりから帯電しやすいプラスチック類は遠ざけてください。
- リストストラップおよび導電性シートを用いない状態で保守部品に触れないようにしてください。

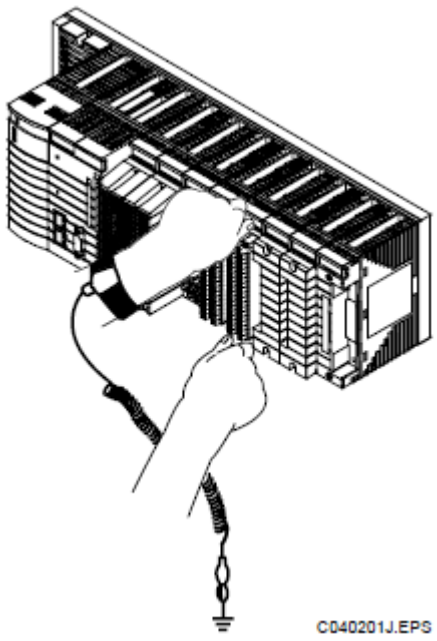


図　モジュールの取り扱い例

● 感電に対する注意事項

- 電源モジュールを交換する場合は、必ず対象となる機器の供給電源を遮断してください。
- 100VAC、200VAC の電圧で信号線を使用する場合、配線時にはフィールド側の電源を OFF にしてください。

● 爆発に対する注意事項

- 爆発性雰囲気の中でない事実が確認されない限り、電源投入状態で作業してはいけません。

B6.4.2 FCN 各モジュールの交換

FCN の各モジュールの交換手順について説明します。FCN では CPU モジュール、電源モジュール、SB バスリピートモジュールを二重化できます。この二重化されたモジュールは、オンラインでハードウェアの交換が可能です。

● 電源モジュールの交換

以下の手順で、電源モジュールを交換してください。二重化構成の電源モジュールは、システムを停止させずに交換できます（シングル構成でも手順は同じです）。

1. 交換対象の電源モジュールへの供給電源を遮断します。
2. 電源ケーブルとラインフィルタ接地ケーブルを端子台から外します。
3. 交換対象の電源モジュールをベースモジュールから外します。
4. 上記の逆の手順で、新規の電源モジュールを取り付け、電源ケーブル、ラインフィルタ接地ケーブルを配線します。
5. 供給電源を投入します。

● CPU モジュール（シングル）の交換

以下の手順で、CPU モジュール（シングル）を交換してください。

1. 交換対象 CPU モジュールが動作中の場合は、シャットダウンスイッチで、停止させてください。
2. 必要に応じて電源を切ってください。（電源を切らずに CPU モジュールを交換した場合、I/O モジュールはフォールバック指定に従った動作をします。）
3. 制御ネットワークケーブルを取り外してください。
4. CPU モジュールをベースモジュールから取り外してください。
5. 取り外した CPU モジュールから FCN/FCJ システムカードを取り出してください。
6. 取り出した FCN/FCJ システムカードを、交換する CPU モジュールへ挿入してください。
7. この CPU モジュールをベースモジュールにネジ止めし、制御ネットワークを取り付けてください。
8. 電源を投入します。

重 要

必ず FCN/FCJ システムカードを実装してから、FCN/FCJ の電源を入れてください。電源が投入されている状態では、FCN/FCJ システムカードを挿抜しないでください。

● CPU モジュール（二重化）の交換

1. 交換するモジュールの制御ネットワークなどの配線を外してください。
2. CPU モジュールをベースモジュールから取り外してください。
3. 取り外した CPU モジュールより、FCN/FCJ システムカードを取り外し、装着する CPU モジュールに、この FCN/FCJ システムカードを取り付けてください。
4. CPU モジュールをベースモジュールにネジ止めしてください。
5. 装着した CPU モジュールがロード完了状態（HRDY LED:点灯、RDY LED:高速点滅、CTRL LED:消灯）なっているのを確認してください。この状態にならない場合は、リセットスイッチで、交換した CPU モジュールをリセットして、ロード完了状態になるのを確認してください。
6. リソースコンフィギュレータあるいはメンテナンスホームページを使用して「B7.2.2　All Program Copy（APC）」「●APC を実行する前の確認事項」に示した項目を確認してください。
7. 最後に制御ネットワークを配線してください。この際、制御ネットワーク 1 あるいは制御ネットワーク 2 のどちらかが制御側 CPU モジュールと通信可能な状態になっている必要があります。通信不可能な場合、APC に失敗し、交換した CPU がダウンします。
8. “APC の自動起動”が有効になっている場合、7.の操作により APC が自動的に実行されます。APC の開始（制御側 CPU モジュール HRDY LED:高速点滅、RDY LED:点灯、CTRL LED:点灯）および完了（待機側 CPU モジュール HRDY LED:点灯、RDY LED:点灯、CTRL LED:消灯）を LED の状態で確認してください。
   “APC の自動起動”が無効になっている場合は、リソースコンフィギュレータより「APC 指令」（All Program Copy）を実行してください。

重　要

APC 実行中は、1 周期のみ制御周期の長さが 1～2 秒程度延びます。また、APC 自動起動設定時の APC 実行のタイミングは、上記のように一過性の CPU 異常の場合など、オペレータの意図しないタイミングで実行される場合があります。そのため、APC 自動起動は、APC 処理がどのタイミングで発生しても問題ないアプリケーションでのみ、設定してください。

参照

CPU モジュール（FCN）二重化時の動作については、「B7.2 CPU モジュール（FCN）二重化時の動作」を参照してください。

● I/O モジュールの交換

以下の手順で、I/O モジュールをオンラインで交換してください。

1. 押し締め端子台または MIL コネクタケーブルを取り外してください。
2. 故障した I/O モジュールを取り外してください。
3. ハードウェア設定している I/O モジュールは、以前と同じ様に設定してください。
4. 新規の I/O モジュールを装着してください。
5. 外した押し締め端子台または MIL コネクタケーブルを装着してください。
6. リソースコンフィギュレータにより、I/O モジュールの設定情報をダウンロードしてください。

● SB バスリピートモジュールの交換

以下の手順で、SB バスリピートモジュールを交換してください。

1. 交換対象の SB バスリピートモジュールを外してください。
2. 分岐コネクタを SB バスリピートモジュール本体より外してください。ケーブルの接続はそのままです。
3. 新規の SB バスリピートモジュール本体に、分岐コネクタを取り付けてください。
4. SB バスリピートモジュールをベースモジュールへ取り付けてください。

重 要

SB バスリピートモジュールは、SB バス延長ケーブルを分岐栓に取り付けたまま交換します（ただし、SB バスケーブルの交換、分岐栓自身を交換する場合を除きます）。この場合、SB バスで接続された他の SB バスリピートモジュールは、影響されずに動作できます。

B6.4.3 FCJ 故障時の交換

以下の手順で、FCJ を交換してください。

1. **FCJ の電源を切ってください。**
2. **FCJ の電源ケーブル、制御ネットワークケーブルを外してください。**
3. **FCN/FCJ システムカードを外してください。**
4. **FCJ のベースボードに本体部を止めているネジ（4 本）を外してください。**
5. **ベースボードから本体部を外してください。**
6. **新規の本体部を、取り外した逆の手順で装着してください。**

補足

FCJ の交換は、本体とベースボードを分離し、フィールド配線をやり直さずに、FCJ 本体を交換できる構成になっています。

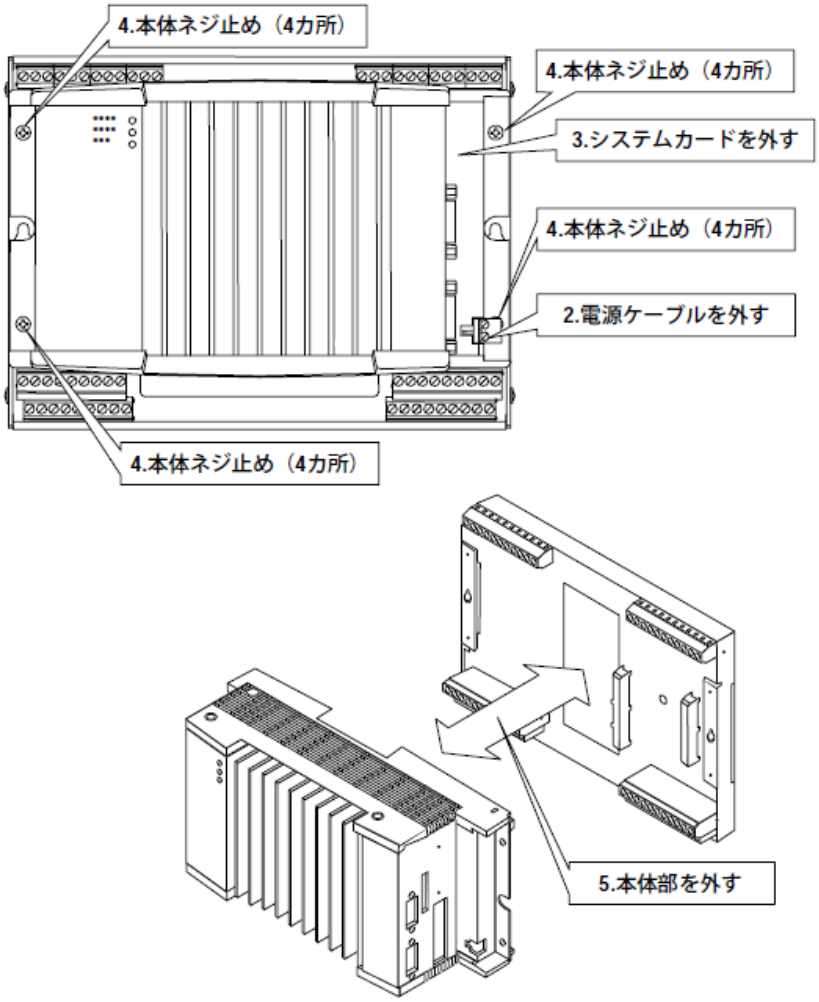


図　FCJ 本体の交換

重 要

FCJ 本体部分の脱着時は、本体部分を落とさないように注意してください。

B6.5 有寿命部品

FCN／FCJ の有寿命部品について説明します。有寿命部品は、使用経過時間とともにその性能や信頼性が低下し、機能を損失し、最終的には故障にいたります。FCN/FCJ では、「モジュール下部吸気温度が平均 45℃の場合、10 年以内に、機能損失・摩耗故障期がくると予測される部品」と定義します。

B6.5.1 FCN CPU モジュール／FCJ のバッテリ

FCN CPU モジュール／FCJ には、電源 OFF 時におけるデータ保持用メモリと RTC（リアルタイムクロック）のために、リチウム（Li）一次電池が搭載されています。電池の推奨交換時期は、使用環境、状態（稼動時間、周囲温度）により異なります。

表　有寿命部品（バッテリ）

品名	部品番号
バッテリ	A1120EB

■ 推奨交換周期

バッテリの寿命は主に周囲温度により決まります。以下に周囲温度と推奨交換周期を示します。本体にバッテリ交換時期を記載可能なラベルが付属していますので、購入時またはバッテリ交換時に、次回のバッテリ交換時期の目安となる年月を記載して目立つ所に貼ってください。

表　バッテリの周囲温度と推奨交換周期

周囲温度 (*1)	交換周期
25℃	10 年
35℃	6 年
45℃	4 年

*1:　周囲温度は使用環境の平均温度を示します。

図　バッテリ交換時期明示用ラベル

重 要

- 周囲温度 25℃以下で使用する場合でも、最長 10 年で交換してください。
- 推奨交換周期を超えて使用すると、電池が液漏れを起こす恐れがあります。
- 液漏れが発生すると、刺激臭発生や金属腐食の恐れがあります。
- 液漏れを起こした電池は速やかに交換してください。
- バッテリを交換するとデータ保持用メモリの内容や設定時刻が失われます。交換作業を依頼する前に、ツールを使用して、データ（リテインデータなど）をバックアップしてください。また、最初に立ち上げたときに時刻を再設定してください。

参照

データのバックアップ方法については、「STARDOM FCN/FCJ 解説書」（IM 34P02Q01-01）「B1.12.1 インタフェースフラグの機能」を参照してください。

■ FCN CPU モジュールのバッテリ

FCN CPU モジュールのバッテリは、CPU モジュールの背面側に実装されています。

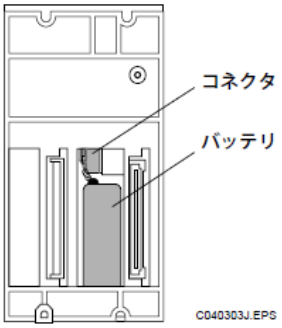


図　バッテリの実装例（FCN）

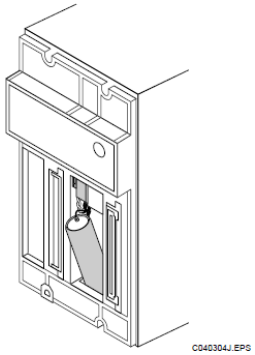


図　バッテリの上側が取り出されている例（FCN）

参照

FCN CPU モジュールのバッテリ交換方法については、「STARDOM FCN/FCJ 解説書」（IM 34P02Q01-01）「C4.3.1 FCN CPU モジュール／FCJ のバッテリ」を参照してください。

■ FCJ のバッテリ

FCJ のバッテリは、FCJ の背面側に実装されています。

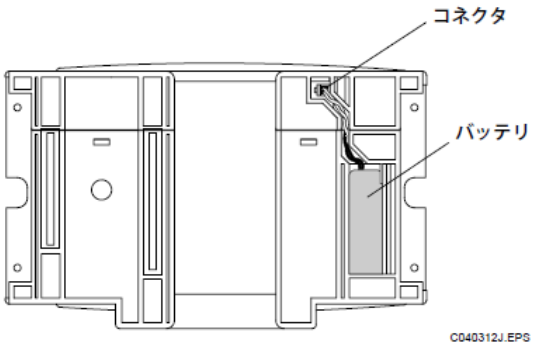


図　バッテリの実装例（FCJ）

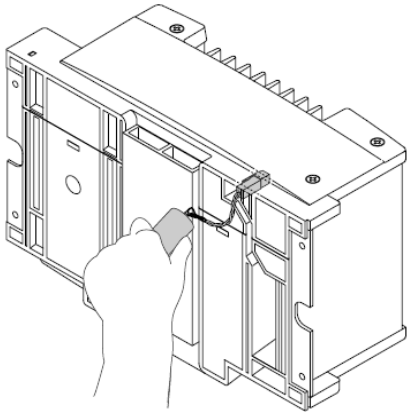


図　バッテリを取り出している例（FCJ）

参照

FCJ のバッテリ交換方法については、「STARDOM FCN/FCJ 解説書」（IM 34P02Q01-01）「C4.3.1 FCN CPU モジュール／FCJ のバッテリ」を参照してください。

B6.5.2 FCN 電源モジュール

電源モジュールで使用している電解コンデンサの寿命は、モジュール下部吸気温度が平均 45℃の場合で 8 年です。定期的に電源モジュールの交換が必要となります。

B6.5.3 FCN I/O モジュール（NFDR541）

I/O モジュールの NFDR541 に使用されているリレーには寿命があり、使用回数、負荷条件が影響します。以下の表を参考にして交換周期を定める必要があります。

補足

リレーの部品交換はできません。モジュールの単体交換になります。

表　リレーの寿命

リレー寿命	機械的寿命	2000 万回		
	電気的寿命	抵抗負荷	AC220 V：2 A	25 万回
			AC100 V：2 A	37 万回
			DC24 V：2 A	27 万回
			DC110 V：0.4 A	25 万回
		誘導負荷	AC220 V：1 A	22 万回
			AC100 V：2 A	16 万回
			DC24 V：0.6 A	30 万回
			DC110 V：0.1 A	35 万回

B7. その他

B7.1 ライセンス

STARDOM ソフトウェアライセンス（以下、ライセンス）について概要を説明します。ライセンスは、以下の手順で入手し、設定してください。

1. **STARDOM のソフトウェアライセンスを購入し、オーダ ID シートを入手します。**
2. **Web ブラウザにより、ユーザを登録します（初回のみユーザ登録が必要となります）。**
3. **Web ブラウザにより、オーダ ID シートに記述されているオーダ ID とパスワードを入力して、ライセンスを発行してください（ライセンスには、キーコードとライセンス ID の 2 種類があります）。**
4. **このライセンスは、ソフトウェアインストール時または、ツールにより STARDOM に設定します（対象となるソフトウェアにより、ライセンスの設定方法が異なります）。**

B7.1.1 ライセンスの種類

STARDOM ソフトウェアライセンスには、以下の種類があります。

● キーコード

ソフトウェアインストール時に必要となるライセンスです。以下のものが対象となります。

（例）

- ロジックデザイナ
- FCN/FCJ OPC サーバ for Windows
- 操作シールドパッケージ
- OPC データリンクパッケージ for .NET

インストール時に必要となるライセンスは、インストール操作の過程で当社から発行されたキーコードを入力すると有効となります。

● ライセンス ID

ソフトウェア実行時に必要となるライセンスです。以下のものが対象となります。

（例）

- 自律型コントローラ基本ソフトウェア
- アプリケーションポートフォリオ
- FCN/FCJ シミュレータ
- VDS/ASTMAC

ソフトウェア実行時に必用となるライセンスは、ソフトウェアの実行ごとに実行権が確認されます。実行権としては、専用ツールを使用して、当社から発行されたライセンス ID をあらかじめ登録しておいてください。

B7.1.2 キーコード

キーコードは、ソフトウェアのインストール時にユーザの正規性を確認するためのライセンスで、ASCII 文字列で提供されます。STARDOM では、ソフトウェアの格納されている DVD-ROM とは別にライセンスを購入すると、キーコードの発行を受けられます。

B7.1.3 ライセンス ID

ライセンス ID は、ソフトウェアの実行時ライセンスを登録するために必要となる、ASCII 文字列のコードです。ライセンスファイルは、ライセンス ID が 1 つまたは複数集まったファイルで提供されます。

STARDOM の実行ライセンスは、あらかじめ所定の登録場所に専用のツールを使用して、ライセンス ID を登録すると有効になります。実行ライセンスを必要とするソフトウェアとライセンス ID の登録場所およびライセンス登録ツールについて、下表に説明します。

表　ライセンスの登録場所とライセンス登録ツール

実行時ライセンスが必要なソフトウェア	登録場所	ライセンス登録ツール
自律型コントローラ基本ソフトウェア	コントローラシステムカード	リソースコンフィギュレータ
アプリケーションポートフォリオ	コントローラシステムカード	リソースコンフィギュレータ
FCN/FCJ シミュレータ	ID モジュール	FCN/FCJ シミュレータユーティリティ
VDS/ASTMAC	ID モジュール	ライセンス管理ツール

STARDOM では、ソフトウェアの格納されている DVD-ROM とは別にライセンスを購入すると、ライセンス ID の発行を受けられます。

B7.1.4 ライセンスの発行

STARDOM のライセンスは、以下の手順で発行されます。

参照

ライセンス管理システム上での各種操作に関しては、Help を参照してください。

ここでは、ライセンスを発行するための基本的な操作の流れについて説明します。

■ オーダ ID シート

STARDOM のソフトウェアライセンスを購入すると、紙のオーダ ID シートが納入されます。このオーダ ID シートには、購入したライセンスの一覧とライセンス管理システムへの接続のキーとなるオーダ ID とパスワードが印刷されています。

■ ライセンス管理システム

STARDOM のライセンス管理システムは、Web 環境が利用されています。インターネットに接続された環境で Microsoft Internet Explorer などの Web ブラウザを起動し、オーダ ID シートに印刷された URL を指定してください。ライセンス管理システムの初期ウィンドウが表示されます。

■ ログイン

ライセンス管理システムを利用するためには、あらかじめユーザを登録しておく必要があります。初期ウィンドウ上では、ユーザ登録で発行されたユーザ ID とパスワードを指定して、ユーザ認証（ログイン）します。ユーザを登録していない場合は、以下のユーザ認証で仮ログインしてください。
ユーザ ID：guest
パスワード：stardom_license

■ ユーザ登録

仮ログインした場合、ユーザ登録のメニューウィンドウが表示されます。「新規登録」を選択してください。ユーザ登録では、所定の入力フィールドがありますので、必要事項を入力して、「登録」ボタンをクリックしてください。登録が受け付けられると、顧客 ID が自動的に発行されます（パスワードはユーザの任意設定です）。

■ オーダ指定とライセンス発行

ユーザ認証が済むと、ライセンス発行のメニューウィンドウが表示されます。「ライセンス発行」を選択するとオーダを指定するウィンドウが表示されます。発行するライセンスの種別とオーダ ID シートに記入されたオーダ ID とパスワードを入力してください。

参照

ライセンスの種別により操作方法が異なります。詳細は、ライセンス管理システムの「操作手順」を参照してください。

■ ライセンス

ライセンスの発行が完了すると「キーコード」または「ライセンスファイル」が入手できます。この入手したライセンスを、各ライセンス登録ツールを用いて、セットアップしてください。

参照

FCN/FCJ のライセンス設定手順については、リソースコンフィギュレータのヘルプを参照してください。

■ ログオフ

ライセンス発行システムを終わるときは、必ずログオフしてください。ログオフせずに Web ブラウザを終了すると、その後 30 分間ログインできなくなりますので注意してください。

B7.2 CPU モジュール（FCN）二重化時の動作

FCN で CPU モジュールを二重化した場合の動作を説明します。

- 制御・待機の各 CPU は全く同一のソフトウェアを並列実行します。制御アプリケーションは、I/O の入出力で制御側と待機側を同期させます。I/O の入力は制御側が Read したデータを待機側に等値化します。I/O 出力は制御側が実行します。この際、制御・待機間で出力値の一致をチェックします。
- 制御・待機間で I/O 出力の不一致が生じた場合、制御側のデータで動作します（不一致が発生した事実をログに出力します）。
- 制御側がダウンした場合、瞬時に待機側に処理を移行し、CPU 切り替えの影響を一切受けずに制御を継続します(*1)。
- 二重化構成の FCN は、ファイルおよびメモリを等値化するため立ち上げに約 110 秒かかります。（ただし、制御は継続して実行されます。）
- シングル CPU 構成で動作中のコントローラに対して、CPU モジュールを追加して二重化構成とする場合、CPU 交換後に APC（All Program Copy：等値化処理）を自動起動させられます。APC 中には、1 周期のみ制御周期が 1～2 秒程度延びます。リソースコンフィギュレータから“APC 指令”を実行する方法によって、APC は手動で開始できます。
- 制御権切り替え時、制御ネットワークの通信が一瞬途切れます。

*1: 即検出不可能なハードウェア異常が発生した場合、切り替えに最大 2 秒かかる場合があります。

■ 二重化構成時の動作

CPU の二重化構成時の動作を示します。制御側 CPU と待機側 CPU モジュールは二重化された内部バス（SB バス）を介して接続されています。制御機能は、制御側、待機側でまったく同じプログラムが「同期・等値化機能」により、同期実行されます。制御側 CPU モジュールが I/O モジュールへアクセスし、結果が待機側 CPU に等値化されます。

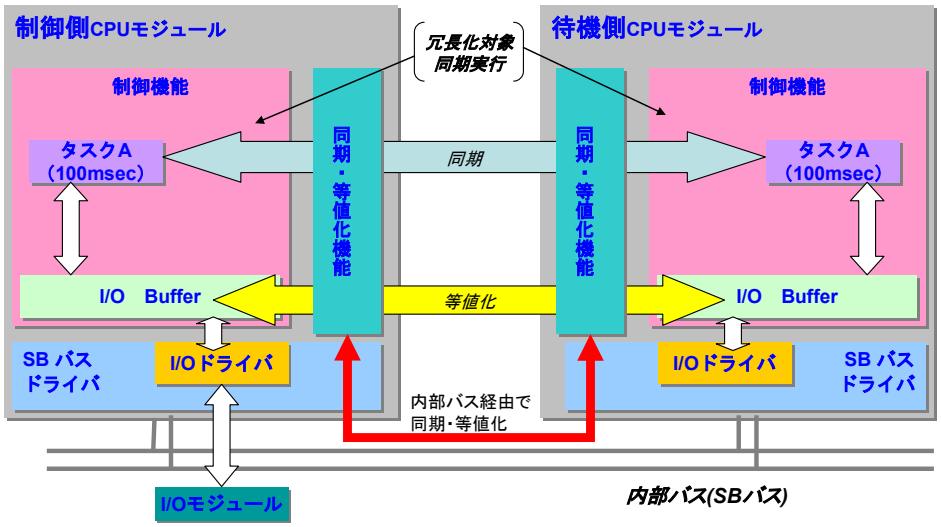


図 二重化 CPU モジュールの内部構造

■ 詳細動作

制御タスクは、入力、制御プログラム実行、出力の順番に定周期で繰り返し実行されます。

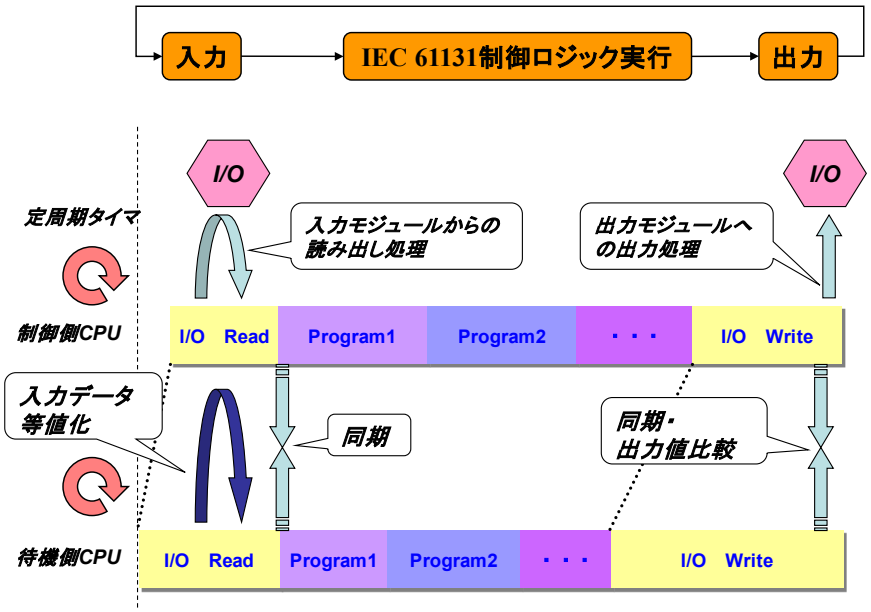


図　二重化 CPU モジュールの等値化

● 通常時の動作

1．入力処理
I/O モジュールからの入力処理を制御側 CPU が実行し、読み出された結果が待機側 CPU に等値化され、双方の CPU が同期されます。

2．プログラム処理
制御側、待機側共に同一のプログラムが実行されます。

3．出力処理
制御側 CPU から出力モジュールに制御出力が書き込まれます。このとき、双方の CPU の出力データが比較され、双方の CPU が同期されます。

● 制御側 CPU 故障発生時の動作

制御側、待機側共に同一のプログラムが実行されます。このとき、制御側 CPU にて致命的な故障が検知されると、制御側 CPU は停止し、同時に待機側 CPU が制御権を持ち、制御側 CPU になります。故障検出後の制御権の移行時間は 100 μS 以下です。

● 同期失敗時の動作

制御側、待機側の CPU が規定時間内に同期されなかった場合、制御側 CPU は同期失敗を記録すると共に、制御を続行します。
待機側の CPU は、停止します。

B7.2.1 CPU モジュール二重化時の制御権

CPU モジュールを二重化した場合、電源投入後、両 CPU モジュールが正常であれば、前回制御を実行していた CPU モジュールが制御側となります。両 CPU モジュールとも初期状態の場合は左側が制御側となります。これは前回の制御権情報を CPU モジュール内のバッテリバックアップ付きメモリに保持しておき、この情報を元に各 CPU モジュールがブート処理において自己診断の後、待ち時間をおいて実現しています。
自己診断処理において前回制御側だった CPU モジュールに異常が検出された場合は、待機側だった CPU モジュールが制御権を持ってシングルとして立ち上がります。一方の CPU モジュールだけを実装して電源を投入し、その後に他方の CPU モジュールを実装した場合は、どちらが制御権を獲得するか不定になります。

表　CPU モジュール立ち上がり時の待ち時間

前回制御権情報	プライオリティ	待ち時間（秒）	
		左側 CPU	右側 CPU
シングル	2	0	2
制御側	1	4	6
待機側	0	8	10

下表「制御権情報の遷移」に制御権を決定するための制御権情報（プライオリティ）の遷移を示します。電源 OFF あるいは異常発生により CPU モジュールがダウンした場合、バッテリバックアップ付きメモリにはダウン前の制御権情報が保持されます。以下の制御権情報のルールにより、一方の異常あるいは不意の電源断が発生してもその後の電源投入時には、以前制御を実行していた方の CPU モジュールが制御側になります。ただし、オフラインで CPU モジュールを交換する場合は、交換する方の制御権情報を初期化（0）する必要があります。

参照

交換する CPU モジュールの制御権情報を初期化（0）する方法については、「B7.2.3 制御権情報の初期化方法」を参照してください。

表　制御権情報の遷移

事象	制御権情報		説明
	左側 CPU	右側 CPU	
二重化構成で初期立ち上げ	1	0	左側が制御側となります。
右側ダウン	2	0	右のダウンを検出し、左はシングル（2）となります。この状態で電源を OFF/ON すると左が制御側となります。
右側復帰	1	0	右の復帰を検出し、左は制御側（1）になります。
左側ダウン	1	2	左の SRAM には 1 が残るが、右は制御権切替を検出してシングル（2）となります。この状態で電源を OFF/ON すると右が制御側となります。
左復帰	0	1	左の復帰を検出し、右は制御（1）、左は待機側（0）となります。

B7.2.2 All Program Copy（APC）

待機側 CPU が制御側 CPU のメモリ内容をコピーする処理（制御側 CPU との等値化）を APC と呼びます。APC の実行は、待機側 CPU の立ち上がりタイミングにより以下の 2 通りの動作があります。

- **起動時 APC**
- **制御実行中 APC**

■ 起動時 APC

起動時 APC は、制御側、待機側 CPU が同時に立ち上がった場合、つまり、二重化構成で電源を投入したときに実行されます。APC 処理は、システムの立ち上げ処理の過程で自動的に実行されます。APC 終了後、制御アプリケーションが起動されます。

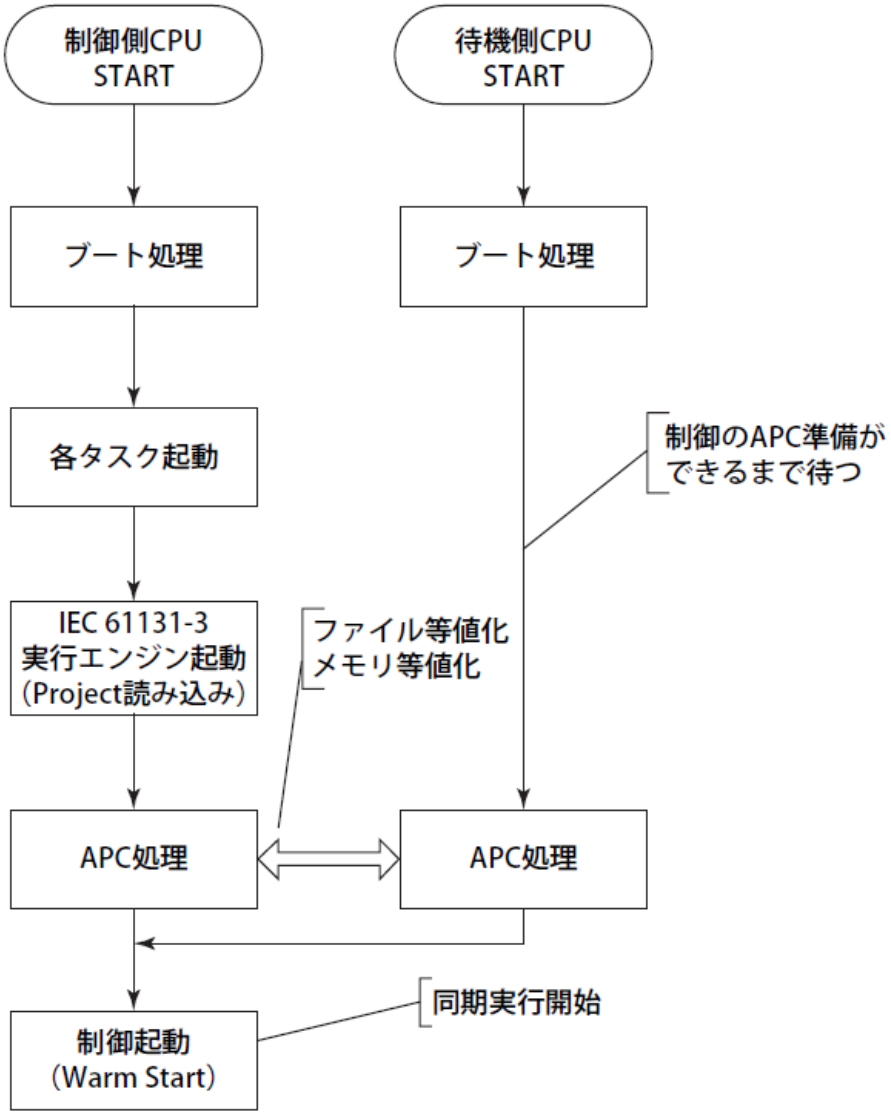


図　起動時の APC

■ 制御実行中 APC

制御実行中 APC は、制御側 CPU が立ち上がっている状態で待機側 CPU を立ち上げた場合、つまり、オンラインで CPU を交換したときなどに実行されます。

● APC 動作

制御実行中 APC の起動は、手動起動と自動起動が選択できます。自動起動の有効／無効は、リソースコンフィギュレータで設定します。デフォルトでは、手動起動です。

表　APC の起動

APC の起動設定	APC 起動条件	APC の動作
手動起動 （自動起動「無効」）	待機側 CPU の交換後、リソースコンフィギュレータの操作で起動	待機側は、ユーザが APC 開始を操作するまで待ち状態となります。リソースコンフィギュレータより制御側に対し、APC 指令を実行すると APC 処理が開始されます。
自動起動 （自動起動「有効」）	待機側 CPU の交換後、LAN ケーブルを接続すると自動起動	待機側では、LAN ケーブルの接続により、自動的に APC 処理が開始されます。CPU モジュール交換作業を簡易化でき、一過性の CPU 異常時に自動復帰できます。

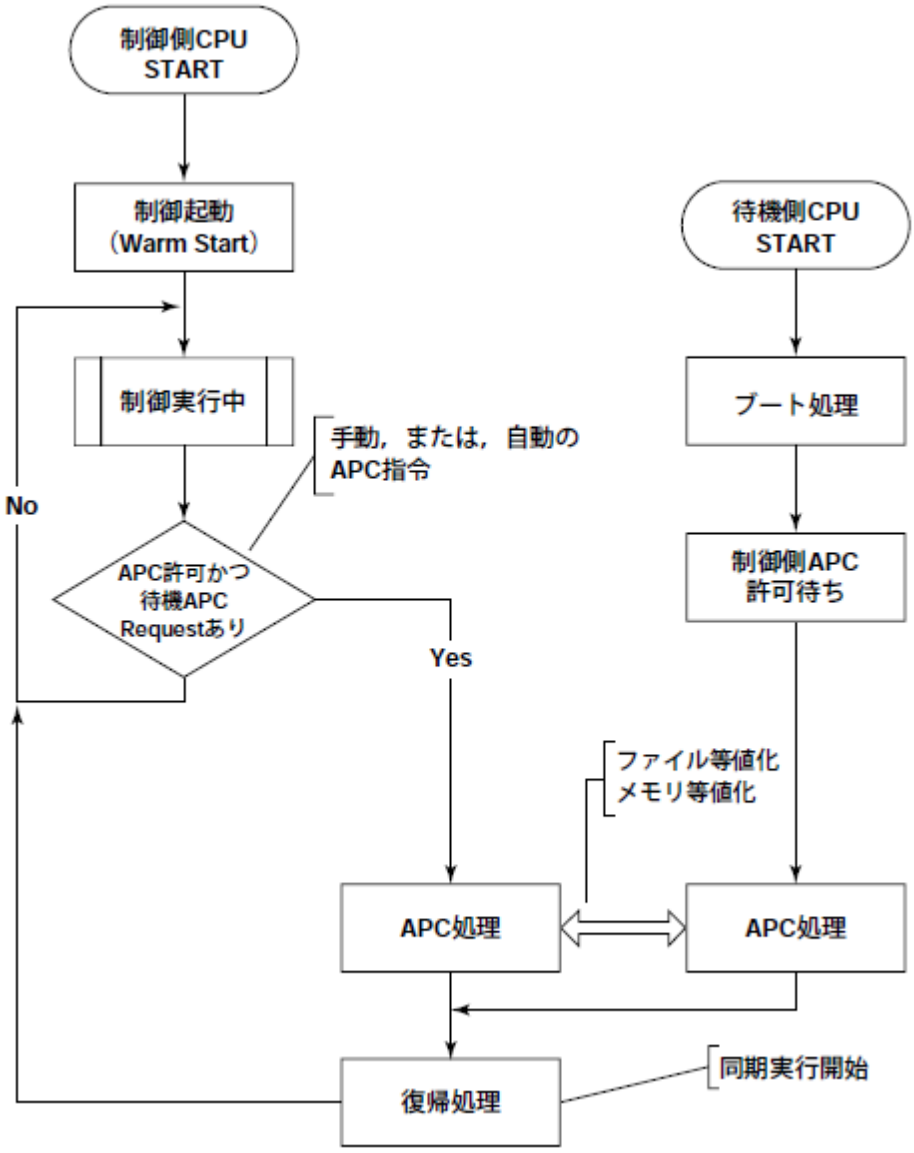


図　制御実行中 APC

待機側 CPU は、手動起動設定時にリソースコンフィギュレータより APC 指令が実行されるか、自動起動設定時に LAN ケーブルが接続されるまで待ち状態となります。この間、制御側 CPU がダウンしても待ち状態のままとなります。この状態で制御側 CPU をリセットした場合は、“起動時 APC”の動作となります（リセットされた方の CPU が制御側となります）。
また、待機側 CPU をシングルあるいは制御側として起動したい場合は、待機側 CPU のシャットダウンスイッチを押して“電源断可能状態”にした後、リセットスイッチを押してください。
APC 実行中には、1 周期のみ、制御周期の長さが 1～2 秒程度延びます。

[APC 自動起動が実行されるケース]

以下のケースで、APC の自動起動が実行され、自動的に CPU 二重化状態に復帰します。

- CPU 二重化状態で動作中の FCN で、一方の CPU に一過性の異常が発生し、CPU 片側ダウン（待機側 CPU 異常、あるいは、制御側 CPU 異常による制御権切り替え）し、すぐ復帰した場合。
- CPU 二重化構成の FCN のシングル CPU 状態で動作中に、待機 CPU をスロットに装着した場合か、待機側 CPU をリセットした場合。

[一過性 CPU 異常の場合の APC 自動起動の動作について]

APC 自動起動設定では、一過性の CPU 異常のときの自動復帰も設定します。その場合は、以下のような動作となります。

- 1 回目の異常発生の場合は、一過性の CPU 異常とみなし、CPU を再起動。
- 1 週間以内に再度異常が発生した場合は、CPU の固定故障とみなし、その CPU を停止（再起動は行わない）。
- 1 回目の異常発生から 1 週間以上正常動作した後に、異常が発生した場合は、一過性の CPU 異常とみなし、CPU を再起動。

重 要

APC 実行中は、1 周期のみ制御周期の長さが 1～2 秒程度延びます。また、APC 自動起動設定時の APC 実行のタイミングは、上記のように一過性の CPU 異常の場合など、オペレータの意図しないタイミングで実行される場合があります。そのため、APC 自動起動は、APC 処理がどのタイミングで発生しても問題ないアプリケーションでのみ、設定してください。

■ APC を実行する前の確認事項

APC を実行する前に必ず以下の点を確認してください。

● APC 自動起動の設定（有効／無効）の確認

リソースコンフィギュレータ、または、メンテナンスホームページより、APC 自動起動の設定が確認できます。
APC 自動起動が有効になっていると、CPU モジュール装着時 (*1) に自動的に APC が実行され、1 周期のみ制御周期の長さが 1～2 秒程度延びます。制御に影響を与える可能性がありますので、APC 実行前に必ず確認してください。
APC 自動起動を有効（Enable）に設定したにもかかわらず、無効（Disable）と表示される場合は、CPU の組み合わせ条件が APC 自動起動を満たさない可能性があります。

*1: 制御ネットワークなどのケーブルなども合わせて実装してください。

● その他の確認事項

- I/O 切り離し機能が利用されていないのを確認します。
- 制御アプリケーションがデバッグ中でないのを確認します。
- プロジェクトコントロールダイアログを表示していないのを確認します。
- ネットワークで制御側／待機側 CPU が接続されているのを確認します。

重 要

- CPU モジュールがシングル構成の FCN を CPU 二重化構成にする場合は、ベースモジュールのディップスイッチ「CPU モジュール二重化設定（SW2）」を「CPU×2」に設定してから CPU モジュールを実装してください。
- CPU モジュールを二重化する場合には、同一レビジョンの基本ソフトウェアを使用してください。

B7.2.3 制御権情報の初期化方法

保守用の CPU モジュールは、あらかじめ以下に示す方法で制御権情報を初期化しておくと CPU モジュール交換作業を簡略化でき、また制御権の切り替わりも発生せず安全に作業できます。初期化しなかった場合、どちらの CPU モジュールが制御側になるかは規定されません。

1. シャットダウンスイッチによる初期化
   シャットダウンスイッチを押したままの状態で電源の投入あるいはリセットスイッチを押してください。“制御権クリア状態”（HRDY LED が低速点滅、他の LED が消灯）になったのを確認してください（約 5 秒でこの状態になります）。この状態にならない場合は、再度本操作を実行してください。
2. 待機側 CPU モジュールの設定による初期化
   シングル構成で動作中の FCN に CPU モジュールをオンライン接続してください。接続した CPU モジュールの立ち上げ処理が実行され、“ロード完了状態”（HRDY LED 点灯、RDY LED 高速点滅、CTRL LED 消灯）になるのを確認してください。この状態にならない場合は、リセットスイッチを押して再度立ち上げ処理を実行してください。“ロード完了状態”になった CPU モジュールは、制御権情報が初期化されています。この後、シャットダウンスイッチを押し、電源断可能状態にして CPU を取り外してください。

B7.2.4 注意事項

● システムソフトウェアのレビジョン

システムソフトウェアのレビジョンは、両 CPU モジュール間で同じにしてください。レビジョンが異なる場合、待機側 CPU から制御側 CPU へのメモリ等値化（APC）要求時にソフトウェアのレビジョンがチェックされ、不一致の場合は待機側 CPU がダウンし、システムアラームが出力されます。

● CPU モジュールのスタイル

制御側および待機側 CPU モジュールのスタイル（ハードウェアのバージョンに相当）は、同一にしてください。

● コンフィグレーション

FCN への IP アドレスなどの設定は、制御側 CPU モジュールに対してのみ可能です。待機側 CPU は“IP アドレス設定状態”および“メンテナンス状態”は、存在しません。待機側 CPU のコンフィグレーション情報は、APC 実行時に等値化されます。

● 等値化処理

待機側 CPU 立ち上がり時、制御側 CPU モジュールの状態を待機側 CPU に等値化するために APC 処理が実行されます。APC は、次の二つのステップで実行されます。この間、リソースコンフィギュレータから FCN へのコンフィグレーション・ダウンロードを実行してはいけません。これらの要求はエラーとなります。下記の処理の実行開始・終了時にシステムアラームメッセージが出力されます。

1. コンフィグレーションファイルの等値化
2. 揮発性メモリおよび不揮発性メモリ（SRAM）内容の等値化
   制御の動作を完全にロックして等値化します。
   「1.」の処理が完了後、制御処理が再開され、IP アドレスなどを格納したシステム設定ファイルが待機側 CPU にコピーされます。等値化するファイルは、システムの最小限のコンフィグレーションファイルのみです。

重 要

待機側 CPU は、「1.」の完了で制御の同期実行を開始しますが、Ready になる（二重化として動作開始する）のは「2.」の完了後です。したがって、「2.」が終了するまではシングル CPU の動作となり、この状態で制御側 CPU がダウンした場合は、システムダウン（両側ダウン）となります。

補足

等値化処理中に制御 CPU が待機側 CPU と通信します。FCN が制御ネットワークに接続されていない場合（スタンドアローンの FCN）でも、等値化処理を実行する時には制御 CPU と待機側 CPU との間をネットワークで接続してください。（クロスのネットワークケーブルでも接続できます。）

● 制御権の切り替え

1. 制御側 CPU に異常が発生した場合
   すみやかに待機側 CPU に制御権が移動し、制御が継続されます。ただし、VDS/ASTMAC や他の FCN/FCJ などと制御ネットワークを介して通信している場合、切り替え処理によって制御ネットワーク通信が一瞬途切れます。その後、復帰します。
2. 待機側 CPU に異常が発生した場合
   制御側 CPU がシングル状態となり処理を継続します。

補足

制御ネットワークの異常を検出しても、制御権は切り替えられません。たとえば、制御側 CPU のネットワークケーブルを外しても、制御権は切り替えられません。

● APC 時の注意事項

APC 実行中には、各機能に以下の影響が発生します。

- ネットワーク二重化機能
  APC 実行中にノードフェールのアラームが発生しますが、APC 完了後すぐに復帰します。
- 制御機能
  APC 実行中に 1 周期のみ制御周期の長さが 1～2 秒程度延びます。そのため、その際に、以下の影響が発生します。

表　APC 実行による各機能への影響

機能	影響
入力処理	影響ありません。(*1)
制御演算	PID などの制御演算が延びます。
出力処理	パルス幅出力、時間比例 ON/OFF 出力が延びます。

*1：PAS ポートフォリオの R1.60.01 以降を使用してください。

B7.3 ネットワークの二重化

VDS/ASTMAC と FCN/FCJ が接続される制御ネットワークは、リアルタイム性と高い信頼性が要求されます。VDS/ASTMAC や FCN/FCJ は、STARDOM ネットワーク二重化機能により、簡単に二重化ネットワークを構築できます。万一片側の通信経路に異常が発生しても、瞬時にかつ自動的に通信経路を切り替えて通信を継続し、アプリケーションへの影響はありません。さらに、FCN の CPU を二重化している場合は、FCN 制御側 CPU に何らかの故障が発生し待機側 CPU と切り替わっても、この FCN と通信をしている VDS/ASTMAC や FCN/FCJ とは、継続して通信できます。

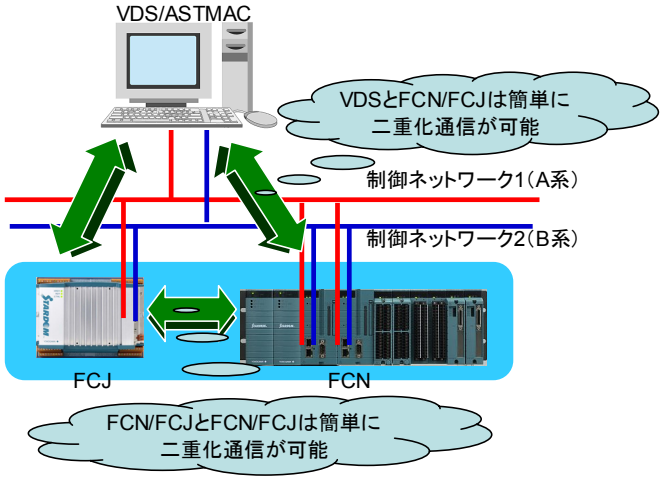


図　STARDOM ネットワーク二重化機能（概念図）

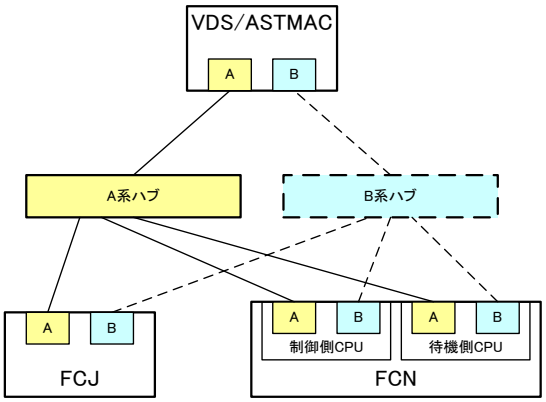


図　STARDOM ネットワーク二重化機能（接続例）

重 要

VDS/ASTMAC と FCN/FCJ が接続される制御ネットワークを二重化する場合、アプリケーションプログラムでは、ネットワークの二重化を考慮する必要はありません。

● 非 STARDOM 機器（PLC など）との通信の二重化

FCN/FCJ は、非 STARDOM 機器（PLC など）との通信を二重化できます。ただし、FCN/FCJ が非 STARDOM 機器との通信を二重化する場合は、アプリケーションプログラムによるエンジニアリングが必要です。

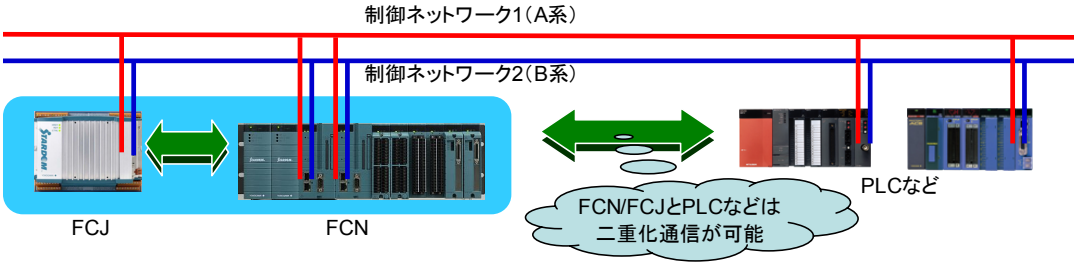


図　FCN/FCJ にネットワークを二重化した PLC を接続した例

重　要

VDS/ASTMAC と非 STARDOM 機器（PLC など）は、一部機器を除き通信の二重化はできません。

参照

- STARDOM ネットワーク二重化機能については、「7.3.1 STARDOM ネットワーク二重化機能」を参照してください。
- 非 STARDOM 機器（PLC など）との通信の二重化については「7.3.2 アプリケーションによる通信の二重化」を参照してください。

■ ネットワーク二重化の手法

STARDOM で提供するネットワーク二重化する方法には、以下の 2 通りがあります。

1. STARDOM ネットワーク二重化機能による通信の二重化
2. アプリケーションによる通信の二重化(*1)

*1: FCN/FCJ システムが R1.85 以上のシステムが対象となります。

表 二重化通信

項目	STARDOM ネットワーク二重化	アプリケーションによるネットワーク二重化
方式	STARDOM ネットワーク二重化機能	アプリケーションプログラムにより実現
二重化通信時に接続可能な機器	・VDS/ASTMAC ⇔ FCN/FCJ ・FCN/FCJ ⇔ FCN/FCJ	・FCN/FCJ ⇔ PLC ・FCN/FCJ ⇔ CENTUM CS3000 FCS ・FCN/FCJ ⇔ FCN/FCJ (他セグメントの場合) (*1)
アプリケーションの対応	不要	必要：アプリケーションで以下を実現 ・アプリケーションで 2 系統のネットワークの異常検出 ・2 系統のネットワークを切り替えます
二重化の範囲	同一セグメント内のみ有効	他セグメント(*1)とも二重化可能
ネットワークの切り替え	コネクションを持続したまま高速に切り替え	コネクションが切れ、アプリケーションによる再接続処理が必要なため、時間がかかります
FCN/FCJ のネットワーク設定	二重化	二重化／独立

*1: ルータ経由の場合などセグメントが分かれている通信

● STARDOM ネットワーク二重化機能

STARDOM では、VDS/ASTMAC⇔FCN/FCJ 間、FCN/FCJ⇔FCN/FCJ 間で、簡単に通信を二重化できます。ただし、この二重化通信機能は、非 STARDOM 機器（PLC など）や、セグメントを越えた通信には適用できません。

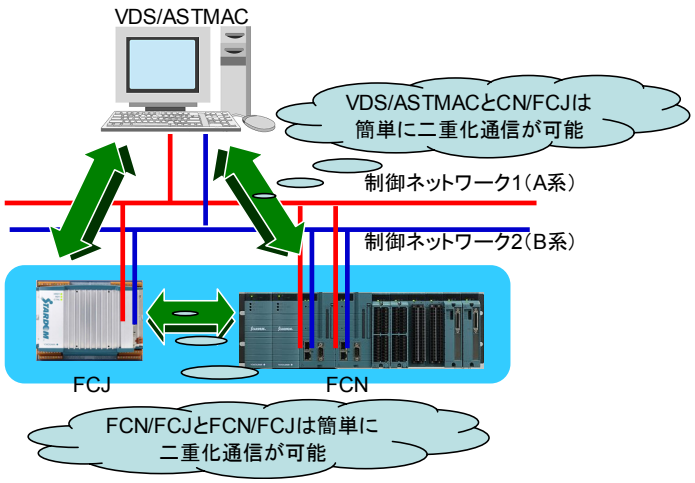


図 STARDOM ネットワーク二重化機能（概念図）

参照

STARDOM ネットワーク二重化機能については、「7.3.1 STARDOM ネットワーク二重化機能」を参照してください。

● アプリケーションによる通信の二重化

FCN/FCJ は、アプリケーションプログラムにより、非 STARDOM 機器（PLC など）や、ルータを経由したセグメント間で、通信を二重化できます。ただし、VDS/ASTMAC は、非 STARDOM 機器（PLC など）や、ルータを経由したセグメント間で、通信を二重化できません。

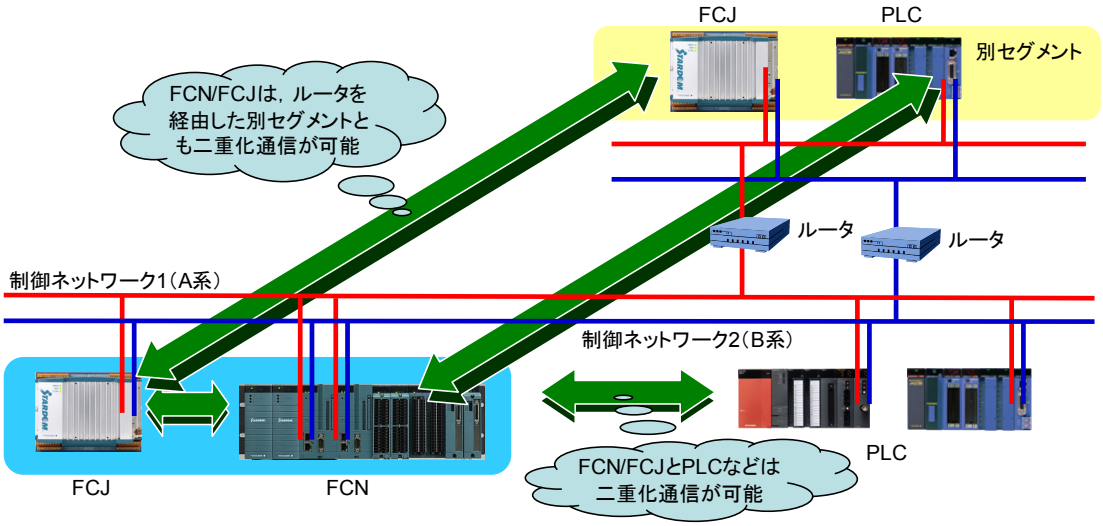


図　FCN/FCJ にネットワークを二重化した PLC を接続した例

参照

アプリケーションによる通信の二重化については、「7.3.2 アプリケーションによる通信の二重化」を参照してください。

B7.3.1 STARDOM ネットワーク二重化機能

STARDOM システムでネットワークを二重化したときの機能を詳細に説明します。

■ ネットワーク二重化の範囲

STARDOM で提供するネットワーク二重化機能は、制御ネットワークの階層で実現され、制御ネットワークのブロードキャストドメインの範囲内で有効です。下図にネットワーク二重化の範囲を示します。

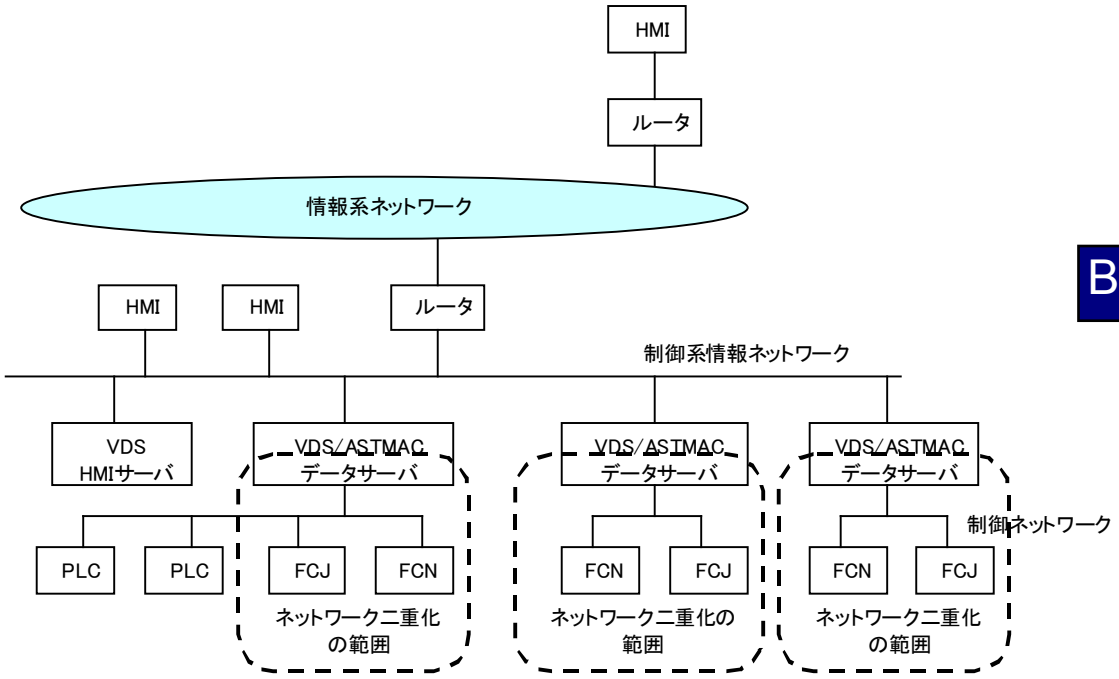


図　STARDOM ネットワーク二重化の範囲

■ 特長

STARDOM の制御ネットワークの二重化は、ネットワークアダプタ、デバイスドライバ、ネットワーク機器（スイッチ、ケーブルなど）のすべてに汎用品を使い、TCP/IP プロトコル上で実現しています。各ノードは複数のネットワークインタフェースを持ち、それらの間の通信経路をルーティングテーブルによって制御しています。通信アプリケーションは、ネットワーク内で唯一のノード名または IP アドレスを使って通信するため、ネットワークの二重化を意識して作成する必要がありません。特長は、ネットワーク経路の切り替え時間が短い、ネットワーク経路の切り替えをアプリケーションが意識する必要がない、などです。また、PLC など、シングル I/F でネットワークの二重化ができないデバイスも、二重化した制御ネットワークに接続できます。

■ 用語の説明

以下に、ネットワーク二重化の構成例を示し、ネットワーク二重化に関する用語について簡単に説明します。

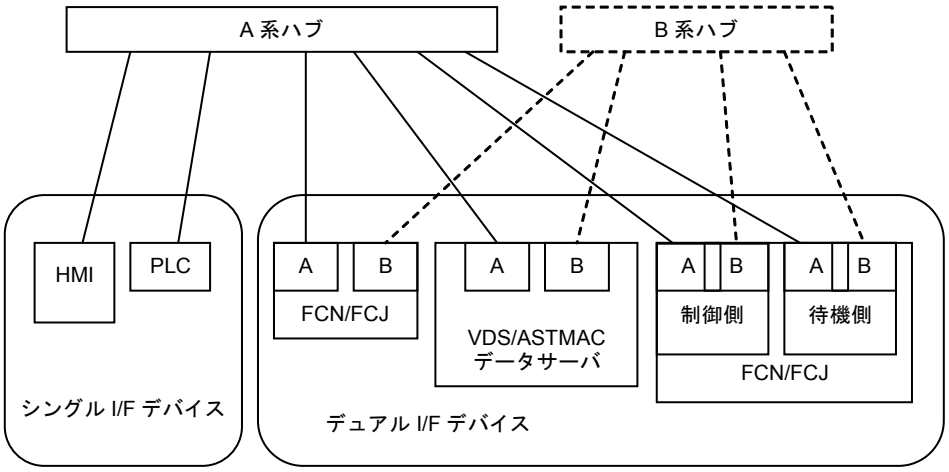


図　二重化ネットワークの基本構成例

●A 系ネットワーク

図中、実線で示したネットワークを A 系ネットワークと呼びます。通常は、こちらのネットワークが使用されます。B 系ネットワークを使用している場合には、B 系ネットワークに異常が検出されると B 系ネットワークの代替として使用されます。

●B 系ネットワーク

図中、点線で示したネットワークを B 系ネットワークと呼びます。A 系ネットワークに異常が検出されると A 系ネットワークの代替として使用されます。

●A 系ハブ

A 系ネットワークに使用するハブを A 系ハブと呼びます。

●B 系ハブ

B 系ネットワークに使用するハブを B 系ハブと呼びます。

●2 つのネットワーク

ネットワークが二重化された STARDOM のネットワークでは、通信するノード間では 2 つの通信経路が存在します。上記 A 系ネットワークと B 系ネットワークです。それぞれのネットワークは対等であり、データ通信にどちらが使用されるかは通信するノードごと（Peer-to-Peer）に決定されます。
したがって、例えばある VDS/ASTMAC に着目した場合、VDS/ASTMAC とあるコントローラ間では A 系ネットワークを使用しているが、他のコントローラとの間では B 系ネットワークを使用している、という状態がありえます。
ただし、あるノードの両方のネットワークケーブルが正しくネットワークに接続された状態でノードが起動されると、一般的には A 系ネットワークを使用します。

● シングル I/F デバイス

制御ネットワーク用のネットワークインタフェースを 1 つしか持たない、または STARDOM ネットワーク二重化機能を実装していないノードです。
シングル I/F デバイスは、A 系ネットワークにのみ接続できます。

● デュアル I/F デバイス

制御ネットワーク用のネットワークインタフェースを 2 つ持ち、STARDOM ネットワーク二重化機能を実装したノードです。ネットワーク二重化機能が有効に設定された FCN/FCJ や VDS/ASTMAC データサーバが対応します。

● インタフェース A

各ノードの 1 番目のネットワークインタフェースをインタフェース A と呼びます。シングル I/F デバイスのネットワークインタフェースはインタフェース A になります。VDS/ASTMAC では、VIP と PIP-A を設定したネットワークがインタフェース A です。FCN/FCJ では、ネットワークインタフェース 1 がインタフェース A です。

● インタフェース B

各ノードの 2 番目のネットワークインタフェースをインタフェース B と呼びます。シングル I/F デバイスには、インタフェース B がありません。VDS/ASTMAC では、PIP-B を設定したネットワークがインタフェース B です。FCN/FCJ では、ネットワークインタフェース 2 がインタフェース B です。

■ IP アドレスの扱い

ネットワーク二重化を実現するために、各ノードに仮想 IP アドレスと物理 IP アドレスを設定する必要があります。アプリケーションは、仮想 IP アドレスだけを意識すれば良いようになっています。物理 IP アドレスは、ネットワーク二重化機能による通信経路のメンテナンスだけに使用します。ネットワーク二重化機能を使わない場合は、仮想 IP アドレスと物理 IP アドレスの考え方は不要になるため、仮想 IP アドレスを IP アドレスに置き換えて理解してください。

● 仮想 IP アドレス（VIP:Virtual IP Address）

ノードに定義された代表 IP アドレスです。リソースコンフィギュレータなどで定義される IP アドレスは、この IP アドレスです。通信アプリケーションはノード名または VIP を使用します。VIP は、PIP-A を設定したネットワークインタフェースに対して設定されます。

補足

CPU を二重化した FCN においては、CPU の制御権が切り換わると制御権と共に VIP も元制御側から新制御側に移行します。
このためアプリケーションは、FCN の CPU 二重化を考慮する必要はありません。

● 物理 IP アドレス（PIP:Physical IP Address）

ネットワークインタフェースに設定された IP アドレスです。VIP をもとに、予め定義されたルール（後述）に従って自動的に生成される IP アドレスです。

● 物理 IP アドレス A（PIP-A）

インタフェース A 側のインタフェースに設定された IP アドレスです。各ノードの PIP-A は、同一のネットワークアドレスになるように設定します。アプリケーションは、このアドレスを意識しません。

● 物理 IP アドレス B（PIP-B）

インタフェース B 側のインタフェースに設定された IP アドレスです。各ノードの PIP-B は、同一のネットワークアドレスになるように、かつ PIP-A と異なるネットワークアドレスになるよう設定されなければなりません。アプリケーションは、このアドレスを意識しません。

■ IP アドレスの決定ルール

FCN/FCJ のネットワーク二重化時の物理 IP アドレス（PIP-A、PIP-B）や FCN で CPU 二重化時の待機側 CPU の IP アドレスは、ユーザが FCN/FCJ に設定する仮想 IP アドレス（VIP）により、一定のルールにより決定されます。
以下にその IP 算出のルールを説明します。

● IP アドレス決定の前提知識

IP アドレスは、通常 192.168.1.1 のように 10 進数の数字 4 つで表されますが、内部処理としては、32 ビットの 2 進数として扱われています。（192.168.1.1 の表記は、上位から 8 ビットづつ 4 つに分け、それぞれ 10 進数に変換してドット（.）でつないで表記しています。上記アドレスを 16 進数で表記すると C0A80101 となります。）
また、IP アドレスは、サブネットマスクによってネットワークアドレスとホストアドレスに分けられます。
サブネットマスクは、255.255.255.0 と設定されますが、ネットワークアドレスは、IP アドレスとサブネットマスクを各々2 進数にして、各ビットの AND を取った値により決定します。（IP アドレス 192.168.1.1 で、サブネットマスク 255.255.255.0 では、16 進数にすると、C0A80101 と FFFFFF00 となり、ネットワークアドレスは、C0A80100=192.168.1.0 となります。）
ホストアドレスは、ネットワークアドレス以下のアドレスとなります。（IP アドレス 192.168.1.1 で、サブネットマスク 255.255.255.0 では、16 進数で 01、10 進数で 1 となります。）
FCN/FCJ のネットワーク二重化時の物理 IP アドレス（PIP-A、PIP-B）や CPU 二重化時の 2 つの CPU の IP アドレスは、仮想 IP アドレス（VIP）のネットワークアドレスとホストアドレスにより、以下の演算を実行すると決定されます。

補足

IP アドレスの先頭に 0 を付けないでください。0 を付けると 8 進数として扱われます。エラーになったり、異なるアドレスになります。Windows 7 以降、厳密にチェックされます。
（例 192.168.01.08 → エラーになる。192.168.01.011 → 9 と認識される。）

● FCN/FCJ CPU シングル、ネットワーク二重化時の PIP 決定ルール

FCN/FCJ で CPU シングル、ネットワーク二重化時の場合は、FCN/FCJ の物理 IP アドレス（PIP-A、PIP-B）は、仮想 IP アドレス（VIP）を元に以下のルールにより決定されます。【PIP-A】

・ネットワークアドレス：VIP のネットワークアドレスに 1 を加えた値
・ホストアドレス：VIP のホストアドレスそのままの値

【PIP-B】

・ネットワークアドレス：VIP のネットワークアドレスに 2 を加えた値
・ホストアドレス：VIP のホストアドレスそのままの値

上記ルールを VIP 192.168.0.1、サブネットマスク 255.255.255.0 の場合に当てはめると、16 進数では、ネットワークアドレスが PIP-A では C0A80100、PIP-B では、C0A80200 で、ホストアドレスが 16 進数で PIP-A、PIP-B とも 01 となります。
これを IP アドレスにすると以下の表のようになります。

表　FCN/FCJ のネットワーク二重化時の IP アドレス

	IP アドレス（16 進数）
VIP	192.168.0.1（C0A80001）
PIP-A	192.168.1.1（C0A80101）
PIP-B	192.168.2.1（C0A80201）

補足

- ネットワーク二重化構成の場合は、IP アドレスとして IPv4 のクラス C のプライベートアドレスを使用し、サブネットマスクには 255.255.255.0 を設定してください。
- 二重化ネットワーク構成の場合は、リソースコンフィギュレータで PIP-A/B を変更できませんので、VIP には、上記ルールで PIP-A/B に矛盾が起こらない値を設定してください。

● FCN CPU 二重化、ネットワークシングル時の IP アドレス決定ルール

FCN で CPU 二重化、ネットワークシングル構成、または、独立ネットワーク構成の場合は、待機側 CPU の IP アドレスは、制御側 CPU の IP アドレスを元に以下のルールにより決定されます。

・ネットワークアドレス：制御側 CPU のネットワークアドレスそのままの値
・ホストアドレス：制御側 CPU のホストアドレスの最上位ビットを 1 とした値

上記ルールをネットワークシングルで制御側 CPU の IP アドレスを 192.168.1.1、サブネットマスク 255.255.255.0 の場合に当てはめると、待機側 CPU の IP アドレスは、16 進数では、ネットワークアドレスは C0A80100 のままで、ホストアドレスは 81（2 進数で 10000001）となります。
これを IP アドレスにすると待機側 CPU の IP アドレスは、C0A80181＝192.168.1.129 となります。
以下の表に、制御側／待機側の CPU の IP アドレスをまとめます。

表　制御側／待機側の CPU の IP アドレス

制御側 CPU の IP アドレス（16 進数）	待機側 CPU の IP アドレス（16 進数）
192.168.1.1（C0A80101）	192.168.1.129（C0A80181）

本構成の場合は、二重化された 2 つの CPU は制御側／待機側に変わるごとに IP アドレスが入れ替わります。

補足

- ネットワークシングル、または、独立ネットワーク構成時には、仮想 IP アドレス（VIP）と物理 IP アドレス（PIP）の区別はなく、単に制御側／待機側の IP アドレスとして扱われます。
- FCN で CPU 二重化時の待機側 CPU の IP アドレスは、上記のように制御側 IP アドレスのホストアドレスの最上位ビットを 1 とした値になります。そのため、制御側 CPU の IP アドレスは、ホストアドレスの最上位ビットが 0 の値を使用してください。たとえば、サブネットマスクが 255.255.255.0 の場合は、制御側 CPU のホストアドレスを 1～126（16 進数で 01～7E）を使用すれば、待機側 CPU のホストアドレスは、129～254（16 進数で 81～FE）となります。
- シングルネットワーク構成や独立ネットワーク構成の場合は、上記ルールによって決定された待機側 CPU の IP アドレスをリソースコンフィギュレータより変更できます。

● FCN CPU 二重化、ネットワーク二重化時の IP アドレス決定ルール（FCN/FCJ：R1.85 以降のシステム）

FCN で CPU 二重化、ネットワーク二重化時の制御側／待機側の CPU の PIP-A/B は、FCN の VIP を元に以下のルールにより決定されます。

【制御側 CPU の PIP-A】
- ネットワークアドレス：VIP のネットワークアドレスに 1 を加えた値
- ホストアドレス：VIP のホストアドレスそのままの値

【制御側 CPU の PIP-B】
- ネットワークアドレス：VIP のネットワークアドレスに 2 を加えた値
- ホストアドレス：VIP のホストアドレスそのままの値

【待機側 CPU の PIP-A】
・ ネットワークアドレス：VIP のネットワークアドレスに 1 を加えた値
・ ホストアドレス：VIP のホストアドレスの最上位ビットを 1 とした値
【待機側 CPU の PIP-B】
・ ネットワークアドレス：VIP のネットワークアドレスに 2 を加えた値
・ ホストアドレス：VIP のホストアドレスの最上位ビットを 1 とした値
上記のように待機側 CPU の PIP-A/B のホストアドレスは、VIP のホストアドレスの最上位ビットを 1 とした値となります。これを VIP 192.168.0.1、サブネットマスク 255.255.255.0 の場合（ホストアドレス 16 進数で 01）に当てはめると、待機側 CPU の PIP-A/B のホストアドレスは、16 進数で 81（2 進数で 10000001）となり、10 進数で 129 となります。本ルールを VIP 192.168.0.1、サブネットマスク 255.255.255.0 の場合に当てはめると PIP-A/B は以下の表のように決定されます。

表　制御側／待機側の CPU の IP アドレス

	制御側 CPU の IP アドレス（16 進数）	待機側 CPU の IP アドレス（16 進数）
VIP	192.168.0.1（C0A80001）＜制御側 CPU に割り当てられます＞	
PIP-A	192.168.1.1（C0A80101）	192.168.1.129（C0A80181）
PIP-B	192.168.2.1（C0A80201）	192.168.2.129（C0A80281）

本構成の場合は、二重化された 2 つの CPU では制御側／待機側に変わるごとに VIP および PIP-A/B の割り当てが変わります。

補足

・ ネットワーク二重化構成の場合は、IP アドレスとして IPv4 のクラス C のプライベートアドレスを使用し、サブネットマスクには 255.255.255.0 を設定してください。
・ FCN で CPU 二重化時の待機側 CPU の IP アドレスは、上記のように VIP のホストアドレスの最上位ビットを 1 とした値になります。そのため、VIP は、ホストアドレスの最上位ビットが 0 の値を使用してください。VIP のホストアドレスには 1～126（16 進数で 01～7E）を使用します。待機側 CPU の PIP-A/B のホストアドレスは、129～254（16 進数で 81～FE）となります。
・ 二重化ネットワーク構成の場合は、リソースコンフィギュレータで PIP-A/B を変更できませんので、VIP には、上記ルールで PIP-A/B に矛盾が起こらない値を設定してください。
・ FCN/FCJ：R1.85 以降のシステムでは、制御側 CPU／待機側 CPU により PIP が自動的に算出されます。FCN/FCJ：R1.85 以前のシステムでは、CPU 実装位置（左側／右側）により PIP が自動的に算出されます。

重　要

FCN/FCJ では、R1.80 以前と R1.85 以降の FCN/FCJ システムでは、IP アドレスの割り付けルールが異なっています。詳細は、「STARDOM FCN/FCJ 解説書」(IM 34P02Q01-01)「D2.2.2 制御ネットワーク二重化構成」を参照してください。

■ トポロジーのバリエーション

制御ネットワークを構成する場合、ネットワークトポロジーにいくつかのバリエーションがあります。以下にそれぞれのトポロジーについて簡単に説明し、最後にそれぞれのトポロジーの特徴を比較して、トポロジー選択の目安を示します。

● ネットワークシングル

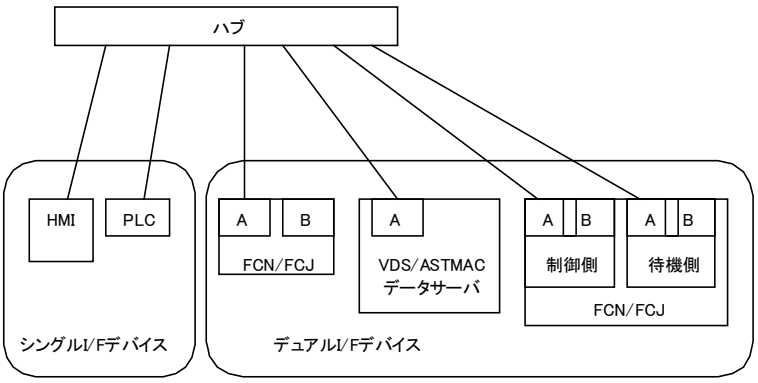


図　ネットワークシングル構成例

ネットワーク経路が二重化されていない構成です。ネットワーク経路が二重化されていないので、ネットワークの故障に対するネットワーク経路の代替がありません。

● ネットワーク二重化

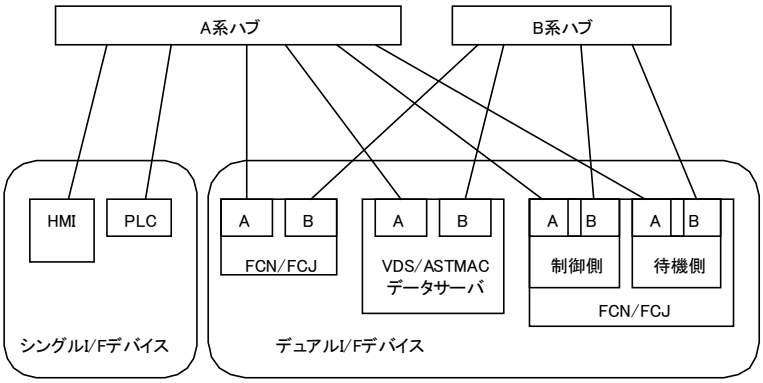


図　ネットワーク二重化構成例

各ノードのネットワークインタフェースが二重化され、それぞれを接続するケーブルおよびハブが完全に二重化され、かつ物理的に独立しているネットワークトポロジーです。フォルトトレランス性は最も高くなります。
シングル I/F デバイスは A 系ハブ側に接続します。シングル I/F デバイスではネットワークが二重化されず、A 系ネットワークでのみ使用できます。

● ネットワークトポロジーの選択の目安

以下に、それぞれのネットワークトポロジーの特徴を表にまとめ、選択の目安を示します。

表　ネットワークトポロジーの比較

	ネットワークシングル	ネットワーク二重化
信頼性	低い	高い（共通部がない）
ネットワーク負荷	低い	低い
コスト（ネットワーク機器）	安い	高い
配線	簡単	やや複雑

ネットワークトポロジーは、お客様のアプリケーションの特性（用途、接続機器、設置場所など）によって最適なものを選択するようにしてください。
FCN/FCJ と VDS/ASTMAC から構成された標準的なネットワーク構成では、信頼性および保守容易性の観点から、ネットワーク二重化の使用を推奨します。

■ ネットワーク二重化の内部動作

STARDOM ネットワーク二重化機能の内部動作について説明します。

● ネットワークステータステーブル（NSTBL）

制御ネットワークに接続される FCN/FCJ および VDS/ASTMAC のネットワーク機能は、各ノードのネットワークインタフェースの状態（ネットワークステータス）を、ネットワークステータステーブル（以下、NSTBL）として管理しています。CPU が二重化されているノードについては、待機側のネットワークインタフェースの情報についても管理します。NSTBL は、すべての VDS/ASTMAC と FCN/FCJ に存在し、ネットワークフェイル、ネットワーク経路切り替え、ネットワークリカバなどのアルゴリズムに使用します。
NSTBL に含まれる情報の例を以下に示します。

表　ネットワークステータステーブルの例

Device Index	VIP	PIP-A	PIP-B	Network 経路情報	
				A 系	B 系
1	192.168.0.1	192.168.1.1	192.168.2.1	OK	OK
2	192.168.0.2	192.168.1.2	192.168.2.2	OK	OK
3		192.168.1.130	192.168.2.130	OK	OK

● 診断通信フレーム

制御ネットワークに接続される FCN/FCJ および VDS/ASTMAC のネットワーク二重化機能は、NSTBL を最新の状態に更新するために、ネットワークインタフェースの状態を診断通信フレームとして定周期にマルチキャスト送信しています。診断通信フレームには、そのノードが管理している NSTBL が格納されています。他ノードから診断通信フレームを受信したノードは、フレームに格納されている NSTBL 情報に従って、自身の NSTBL を最新の状態に更新します。診断通信フレームは、二重化されているそれぞれのインタフェースから送信されます。診断通信周期のデフォルト値は 500ms です。診断通信が 2 回続けて失敗した場合にフェイルとみなすので、診断通信周期 500ms のとき、切り替え時間は 1000～1500ms となります。

診断通信フレームは、二重化されている機器（FCN や VDS/ASTMAC）数の分ネットワーク上に発信され、各機器が受け取ります。診断通信フレームを受け取った機器は、フレームを受け取るごとに処理を実行しますので、二重化されている機器が多くなるごとに診断通信フレーム受信処理が多くなり CPU 負荷が高くなります。そのため、二重化されている機器が多い場合は、診断通信フレームの送信周期を長くしてください。

以下に機器数と設定する診断周期の目安を示します。

表 CPU 数と診断周期の目安

CPU 数	診断周期（msec）	備考
12 まで	500	診断周期デフォルト値
13～25	1、000	
26～37	1、500	
38～50	2、000	
51～62	2、500	
63～75	3、000	
76～87	3、500	
88～100	4、000	
101～112	4、500	
113～125	5、000	
126～137	5、500	
138～150	6、000	
151～162	6、500	
163～175	7、000	
176～187	7、500	
188～200	8、000	
201～212	8、500	
213～225	9、000	
226～237	9、500	
238～248	10、000	

注： CPU 数とは、ネットワーク中のネットワーク二重化機能を使用する FCN/FCJ/VDS/ASTMAC の台数（FCN の CPU 二重化時は 2 とする）の合計です。

● ネットワークフェイル

二重化されたネットワークの片側の経路が使用不可能になった状態をネットワークフェイルと呼びます。ネットワークフェイルはノードごとに管理されています。ある特定のノードから一定時間診断フレームを受信できないとき、そのノードに対するネットワーク経路がフェイルしたと判断され、メッセージとしてオペレータに通知されます。ネットワークフェイルの原因には、自ノードのネットワークインタフェースの故障、相手ノードのネットワークインタフェースの故障、両ノード間のネットワーク経路を構成する機器（ハブ、ケーブル）の故障、一時的なノイズによる通信妨害などが考えられます。

● ネットワーク経路切り替え

あるノードに対するネットワークフェイルが検出されると、自動的に Routing Table を変更し、二重化されたもう 1 つのネットワーク経路へ速やかに切り替えられます。

● ネットワークリカバ

ネットワークフェイルを検出したノードは、フェイルしたネットワーク経路の回復を継続的に監視しています。ネットワーク経路の回復が検出されると、ネットワークリカバのメッセージがオペレータに通知されます。

■ FCN の CPU 二重化対応

FCN の制御側（制御を実行している側）の CPU がフェイルすると、待機側（制御側と常に等値化され制御側がフェイルしたときに制御権を引き継げる状態にある側）の CPU に制御権が移動します。制御側と待機側は、制御ネットワークを経由して等値化されます。制御権を新たに獲得した CPU は、以前制御権をもっていた CPU のネットワーク情報（仮想 IP アドレス、ホスト名、その他）を引き継ぎ、通信を継続します。
この機能により、CPU の切り替えが発生した FCN と通信していた他のノードの通信アプリケーションは、その FCN の CPU の切り替え動作を意識して作成しておく必要がありません。

● ホスト名と IP アドレス

CPU を二重化した FCN のホスト名および IP アドレス（仮想 IP アドレス）は、クライアント側（VDS/ASTMAC など）のアプリケーションから見ると常に不変であり、単一の機器として認識されます。FCN の CPU に切り替えが発生すると、仮想 IP アドレスが切り替わった CPU 側へ瞬時に移動します。

● 待機側 CPU のネットワークインタフェースのステータス

待機側 CPU のネットワークインタフェースの状態についても、常にチェックされ、チェック結果は NSTBL に反映されます

B7.3.2 アプリケーションによる通信の二重化

FCN/FCJ は、アプリケーションプログラムにより、非 STARDOM 機器（PLC など）との通信を二重化できます。
ただし、VDS/ASTMAC は、通信をこの様には二重化できません。
アプリケーションによる通信の二重化により接続可能な機器には、下記があります。

- PLC（FA-M3R、MELSEC など）
- エムシステム　リモート I/O
- 他のセグメントにある FCN/FCJ

など

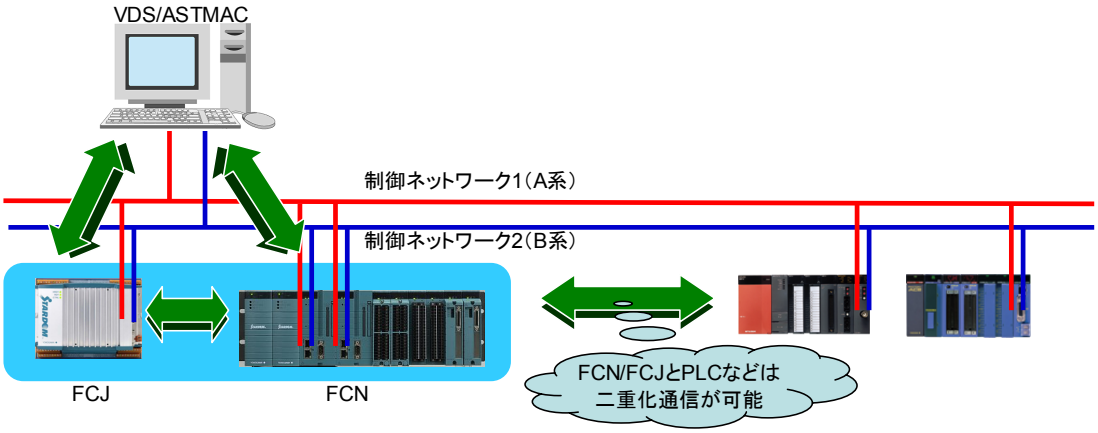


図　FCN/FCJ にネットワークを二重化した PLC を接続した例

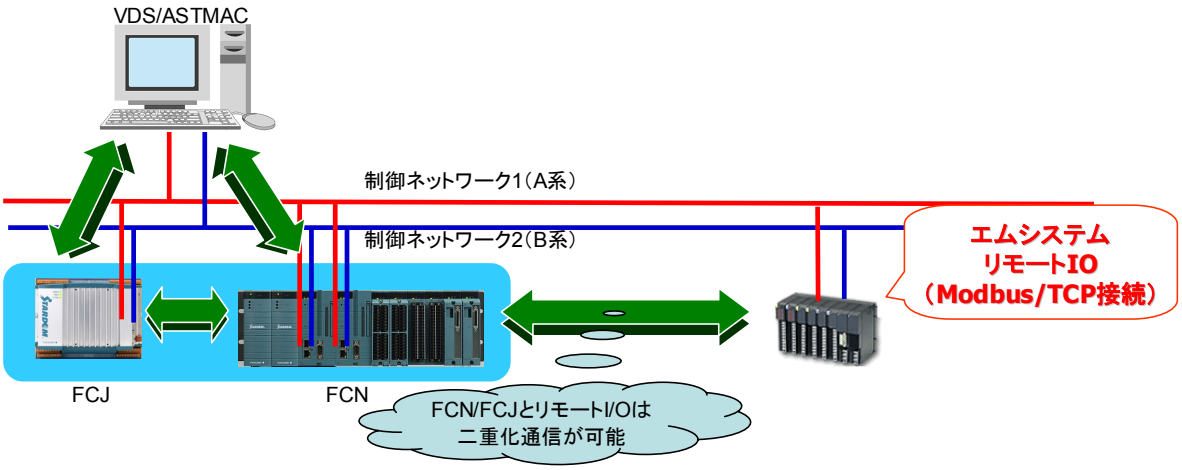


図　FCN/FCJ にネットワークを二重化したリモート I/O を接続した例

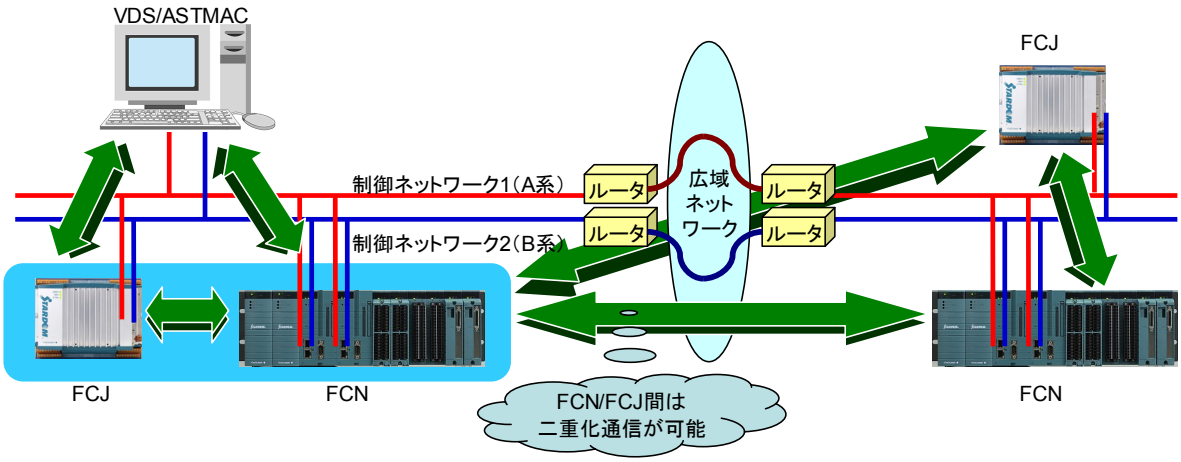


図　遠隔の FCN/FCJ をルータ経由で接続した例

B

■ アプリケーションプログラム

非 STARDOM 機器（PLC など）や、ルータを経由したセグメント間で、通信を二重化する場合、通信アプリケーションが必要です。
通信アプリケーションには、下記のロジックを実装してください。

● アプリケーションに組み込むロジック

1. 通信異常の検出
2. 通信経路の切り替え

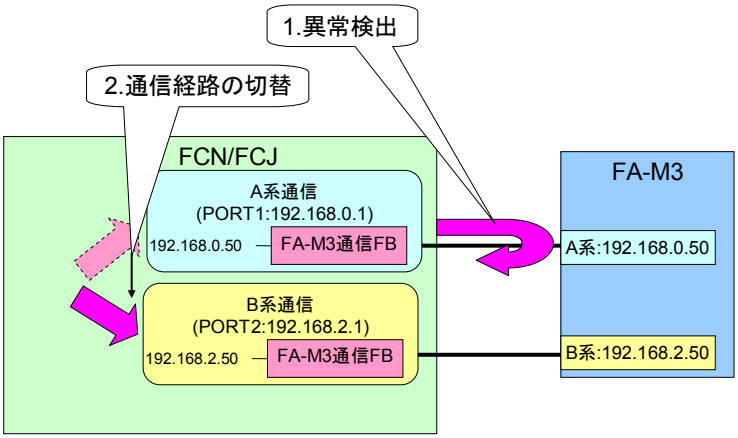


図　アプリケーションに組み込むロジック（FA-M3 と通信した例）

■ 「二重化」設定と「独立」設定

FCN/FCJ がアプリケーションによって通信を二重化するとき、以下の 2 とおりのネットワーク構成が可能です。

- 二重化ネットワーク構成
- 独立ネットワーク構成

STARDOM では、「STARDOM ネットワーク二重化機能」および「アプリケーションによる通信の二重化」が可能な「二重化ネットワーク構成」を推奨します。

B7.4 通停電時の動作

FCN/FCJ の通電時および停電時の動作について説明します。
瞬停、停電の発生記録が残っているので後から解析できます。

B7.4.1 停電不感時間

停電不感時間とは、電源ラインの電圧がある時間だけ一時的に低下したときに、制御に影響しないと保証される最小の時間です。この時間以内であれば、電圧がゼロになっても FCN/FCJ の制御に影響しません。

● FCN の場合

FCN の電源モジュールの停電不感時間の仕様を下表に示します。

表　FCN 電源モジュールの停電不感時間

形名	定格入力電圧	入力電圧下限	停電不感時間
NFPW441	100～120 V AC	80V	10 msec
NFPW442	220～240 V AC	170V	10 msec
NFPW444	24 V DC	21.6V	2 msec

● FCJ の場合

FCJ は FCN と内蔵電源回路の方式が異なるため、停電不感時間はありません。入力電圧 24VDC が 0V に低下すると、即時に停電と判断します。

B7.4.2 瞬停時／停電時の FCN の動作

瞬停とは、停電不感時間内で復帰する停電です。

● 瞬停時の FCN の動作

1. 電源モジュールでは入力電圧の低下を検知し、この状態が停電不感時間を越えると、停電予告信号を CPU モジュールに送ります。
2. CPU モジュールは、この停電予告信号を受けると、直ちにシステムアラームに AC フェイルの発生を記録します。
3. この後、 CPU モジュールは、AC フェイル状態からの復帰をシステムアラームに記録します。

● 停電時の FCN の動作

1. 電源モジュールでは入力電圧の低下を検知し、この状態が停電不感時間を越えると、停電予告信号を CPU モジュールに送ります。
2. CPU モジュールは、この停電予告信号を受けると、直ちにシステムアラームに AC フェイルの発生を記録します。

B7.4.3 停電復帰後の動作

**停電からの復帰時には、通常の電源投入時と同一の動作になります。
チューニングパラメータ（モードを含む）、ユーザ設定の保持指定データなどについては、保持変数用メモリに保持されたデータが使用されます。
ただし、I/O モジュールに接続されている NPAS POU のモードは、強制的に MAN（手動操作状態）に遷移します。**

● 停電復帰後に通常制御に戻るまでの時間

FCN/FCJ が停電後に復旧して通常の制御状態になるまでの時間は、次の和になります。

1. OS 立ち上がり時間：
   CPU モジュール（シングル構成）：約 20 秒
   CPU モジュール（二重化構成）　：約 1 分 50 秒
2. I/O 立ち上がり時間：10 秒
3. 制御ロジック立ち上がり時間：
   スキャン周期の 30 周期分（スキャン周期が 1 秒の場合は、30 秒）

注：　ただし、上記時間は目安です。プロジェクトサイズ等、アプリケーションにより時間は変動します。

B7.5 障害発生時の動作

B7.5.1 自己診断

■ RAS 機能

FCN/FCJ の RAS(*1)機能は、CPU（FCN/FCJ）の自己診断や I/O モジュール（FCN）の状態監視によって、FCN/FCJ の安定動作を支援します。
自己診断機能には、下記のものがあります。

● FCN/FCJ の CPU

- ECC メモリの診断
- 内部バスの診断
- WDT(*2)による診断

など

● FCN 二重化 CPU モジュール

- CPU 間での同期監視
- 出力不一致の監視

など

● FCN I/O モジュール

- アナログ入力回路のオープンやショートの検出
- アナログ出力のリードバック
- 内部 CPU のタイマなどの資源

など

*1: 信頼性（Reliability）、可用性（Availability）、保守性（Serviceability）の略。
*2: ウォッチドッグタイマ（Watch Dog Timer）の略。

B7.5.2 動作状態

FCN/FCJ の動作状態は、下図のように遷移します。それぞれの状態は CPU モジュールの 3 つの LED の状態（下記参照）で判断できます。

1. H（HRDY）点灯は、ハードウェア準備完を意味します。
2. R（RDY）点灯は、ソフトウェア準備完を意味します。
3. C（CTRL）点灯は、コントローラ制御中を意味します。

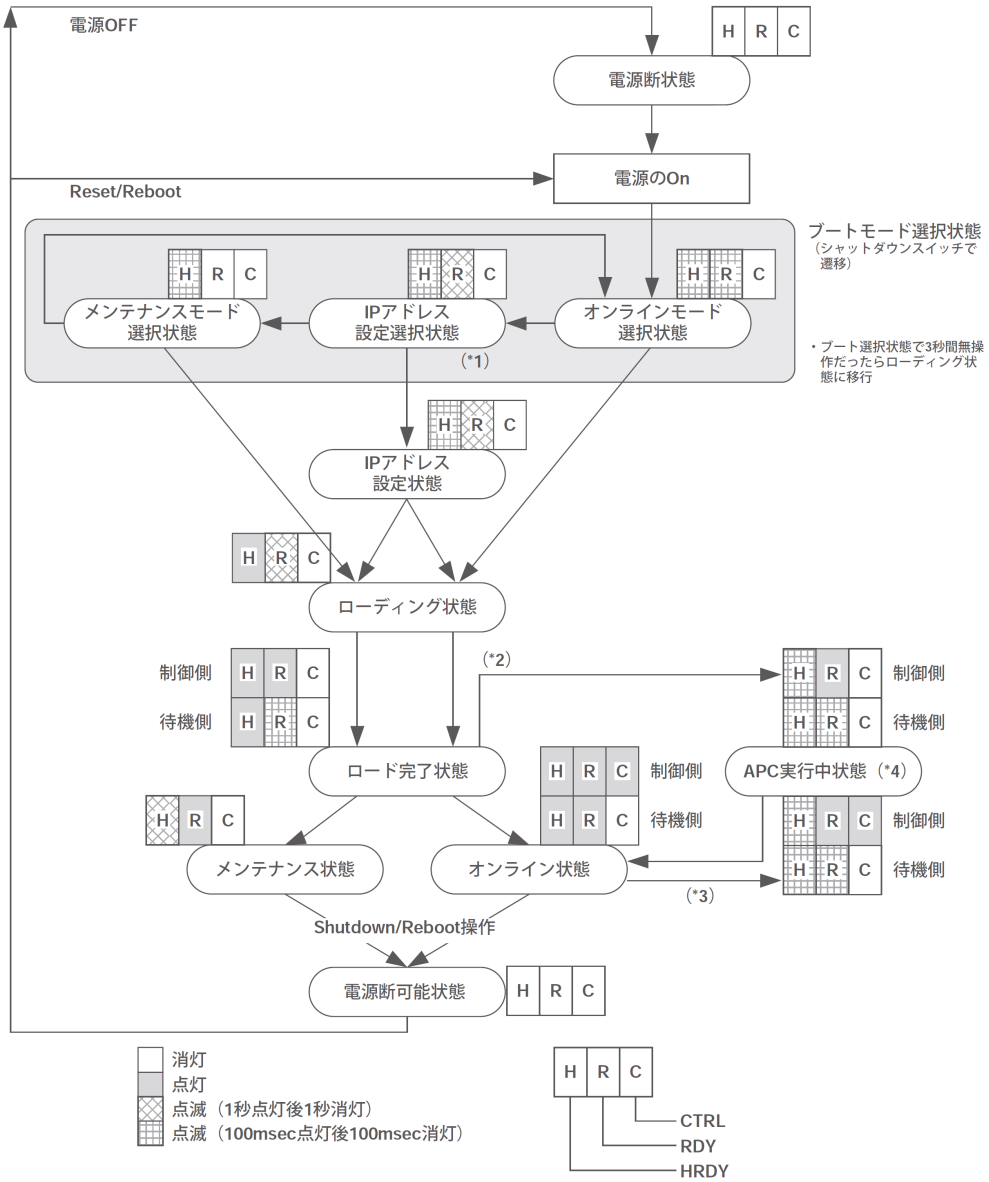


図　FCN/FCJ の動作状態

*1: IP アドレスが設定されていない状態やブートモード選択状態（3 秒）で FCN/FCJ 自律型コントローラのシャットダウンスイッチを一度押すと、「IP アドレス設定選択状態」になります。
*2: CPU モジュールが二重化された FCN の場合のみ、APC 実行中状態になります。
*3: 「APC 指令」が実行されたときに APC 実行中状態になります。
*4: 「APC 実行中状態」の LED 表示は、制御側 CPU と待機側 CPU とでは、同時に変化しません。次表のように待機側 CPU の LED 表示が途中で遷移します。これは APC 処理のステータスを示しています。

表　APC 実行中状態の CPU モジュール LED 表示

APC 処理ステータス	CPU モジュールの LED 表示	
	制御側 CPU	待機側 CPU
1.ファイルのコピー中	APC 実行中状態(*1)	ローディング状態
2.メモリのコピー中	APC 実行中状態(*2)	APC 実行中状態

*1:　制御アプリケーションは、継続して動作します。
*2:　制御アプリケーションは、一時的に停止します。

● ブートモード選択状態

ブート選択状態には、下記の 3 つの状態があります。
ブート選択状態の時にシャットダウンスイッチを押すと、選択状態が順次遷移します。

1. オンラインモード選択状態（初期値）
2. IP アドレス設定選択状態
3. メンテナンスモード選択状態

3 秒間シャットダウンスイッチの操作をしないと選択された状態で起動されます。ただし、IP アドレスが未設定の場合には、常に「IP アドレス設定選択状態」になります。

● IP アドレス設定状態

FCN/FCJ の IP アドレスを設定する状態です。IP アドレスの設定が完了するとローディング状態になります。

● ローディング状態

FCN/FCJ システムカードに格納されたシステムを、メモリ上にローディングする状態です。

● ロード完了状態

ローディングが完了し、OS の起動がかかった状態です。実行モードによりオンライン状態、またはメンテナンス状態になります。

● オンライン状態

通常の動作状態（制御アプリケーションなどが動作できる状態）です。

● メンテナンス状態

最小限の OS 機能が動作している状態です。制御アプリケーションは停止しており、高度なコンフィグレーションやファイルのバックアップ／リストアなどのメンテナンス操作を実行できます。

● APC 実行中状態

FCN の CPU モジュールが二重化されたシステムにおいて、APC（All Program Copy）処理の実行により、制御側 CPU の内容が待機側 CPU にコピーされている状態です。

● 電源断可能状態

コントローラの電源を落としてもよい状態。
電源断状態と電源断可能状態の CPU モジュールの LED 点灯状態は同じですが、電源モジュールの LED で識別してください。

Technical Information　改訂情報

資料名称　　：STARDOM 計装テクニカルガイド
資料番号　　：TI 34P02A03-01

2006 年 06 月／初版
・新規発行

2006 年 10 月／2 版
・FCN/FCJ R1.85 対応
・誤記訂正

2007 年 7 月／3 版
・VDS/ASTMAC R5.50 対応
・FCN/FCJ R1.90 対応
・誤記訂正

2008 年 9 月／4 版
・VDS/ASTMAC R6.01 対応
・FCN/FCJ R2.01 対応
・誤記訂正

2009 年 10 月／5 版
・FCN 用途 NFBU050 対応
・VDS/ASTMAC R6.20 対応
・FCN/FCJ R2.20 対応
・誤記訂正

2010 年 8 月／6 版
・ソフトウェア媒体変更
・VDS/ASTMAC R6.30 対応
・MELSEC L シリーズ Ethernet 接続対応
・FCN/FCJ R3.01 対応
・FCN/FCJ 制御ループ構築機能を削除
・誤記訂正

2011 年 6 月／7 版
・VDS/ASTMAC R7.01 対応
・FCN/FCJ R3.10 対応
・Windows 7 対応
・VDS Viewer 追加
・通信ケーブル情報追加
・誤記訂正

2012 年 8 月／8 版
・VDS/ASTMAC R7.10 対応
・FCN/FCJ R3.20 対応
・通信モジュール情報追加
・.NET アプリケーション追加
・誤記訂正

2013 年 6 月／9 版
・VDS R7.20 対応
・FCN/FCJ R3.30 対応
・ASTMAC を削除
・計装 CPU モジュール追加
・誤記訂正

2014 年 7 月／10 版
・VDS R7.30 対応
・FCN/FCJ R3.40 対応
・Windows 7　64bit 版を 追加
・Windows XP を削除
・適合規格更新

■ お問い合わせについて
本書の内容に関するご質問は、下記メールアドレスにてお願いいたします。
問い合わせメールアドレス：stardom_info@cs.jp.yokogawa.com
■ 著作者　横河電機株式会社
■発行者　横河電機株式会社
〒180-8750　東京都武蔵野市中町 2-9-32